

# **Programová podpora návrhu MIMO řízení polynomiálním přístupem**

Computer-aided Polynomial MIMO Control System Design

Bc. Petr Francálek

---

Diplomová práce  
2013



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2012/2013

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr Francálek**  
Osobní číslo: **A11418**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Automatické řízení a informatika**  
Forma studia: **prezenční**

Téma práce: **Programová podpora návrhu MIMO řízení  
polynomiálním přístupem**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Nastudujte si princip návrhu MIMO řízení polynomiálním přístupem.
3. Navrhněte a realizujte GUI, které umožní navrhout MIMO systém řízení polynomiálním přístupem pro klasickou zpětnovazební konfiguraci, přičemž uvažujte MIMO systémy alespoň do rozměru  $3 \times 3$ .
4. Jako algoritmus řešení implementujte metodu postupných polynomiálních redukcí s využitím elementárních operací na polynomiální matici.
5. Vstupem aplikace budou polynomiální matice řízeného systému, tvar vstupních signálů a požadované chování výsledného regulačního obvodu. Výstupem bude navržený regulátor, který zajistí požadované chování.
6. Umožněte přímé nebo nepřímé ověření výsledku návrhu řízení simulačními prostředky.
7. Na vybraných případech porovnejte získané výsledky s řešením pomocí jiného přístupu.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. DOSTÁL, Petr; MATUŠŮ, Radek. Stavová a algebraická teorie řízení. Zlín: UTB ve Zlíně, Fakulta aplikované informatiky, 2010. ISBN 978-80-7318-991-4.
2. PROKOP, Roman, Radek MATUŠŮ a Zdenka PROKOPOVÁ. Teorie automatického řízení: lineární spojité dynamické systémy. Zlín: UTB ve Zlíně, 2006. ISBN 8073183692.
3. HAVLENA, Vladimír a Jan ŠTECHA. Moderní teorie řízení. Praha: ČVUT, 1996. ISBN 8001010767.
4. ŠEBEK, Michael. Systémy a řízení: 8. přednáška – Polynomiální metody (online). 2007 (cit. 2013-02-04). ČVUT v Praze. Dostupné z: <http://www.avc-cvut.cz/archiv/prednasky/prednasky-fel-cvut/x35sri---systemy-a-řízení>
5. SKOGESTAD, Sigurd; POSTLETHWAITE, Ian. Multivariable Feedback Control: Analysis and Design. Chichester: Wiley, 2005. ISBN 0-470-01168-8.
6. ZAPLATÍLEK, Karel; DOŇAR, Bohuslav. MATLAB – tvorba uživatelských aplikací. Praha: BEN – Technická literatura, 2004. ISBN 80-7300-133-0.
7. KARBAN, Pavel. Výpočty a simulace v programech Matlab a Simulink. Praha: BEN-technická literatura, 2007. ISBN 978-80-251-1448-3.

Vedoucí diplomové práce:

**doc. Ing. František Gazdoš, Ph.D.**

Ústav řízení procesů

Datum zadání diplomové práce:

**24. února 2013**

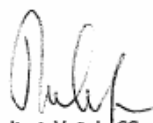
Termín odevzdání diplomové práce:

**11. června 2013**

Ve Zlíně dne 24. února 2013

  
prof. Ing. Vladimír Vašek, CSc.  
děkan



  
prof. Ing. Vladimír Vašek, CSc.  
ředitel ústavu

## **ABSTRAKT**

Tato diplomová práce se zabývá návrhem MIMO řízení polynomiálním přístupem. Cílem je vytvoření programové podpory pro řešení tohoto návrhu. Teoretická část práce se zabývá možnostmi realizace této programové podpory, základními pojmy, vztahy a popisuje polynomiální metodu návrhu řízení mnohorozměrných systémů. Praktická část popisuje použité vývojové prostředky, algoritmus a výslednou aplikaci. Funkčnost je demonstrována na několika ukázkových úlohách.

Klíčová slova: návrh řízení, mnohorozměrné systémy, polynomiální přístup, programová podpora, regulace, polynomiální matice

## **ABSTRACT**

This thesis deals with polynomial MIMO control system design. The aim of thesis is to create a computer-aided solution for this method of control system design. The theoretical part deals with possibilities of realization of computer-aided solution, basic concepts, relationships and polynomial MIMO control system design. The practical part describes used development tools, algorithm and the final application. The functionality is demonstrated on several sample tasks.

Keywords: control system design, MIMO systems, polynomial approach, computer-aided, regulation, polynomial matrices

Děkuji vedoucímu mé diplomové práce panu doc. Ing. Františku Gazdošovi, Ph.D. za pomoc, cenné rady a celkově skvělé vedení práce. Dále děkuji také rodině za podporu v průběhu studia.

Motto:

*„Non scholae, sed vitae discimus“*

*(Neučíme se pro školu, ale pro život)*

Seneca

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 SOUČASNÝ STAV PROGRAMOVÉ PODPORY NÁVRHU MIMO ŘÍZENÍ</b> .....	<b>11</b>
1.1 MATLAB .....	11
1.1.1 Polynomial Toolbox .....	12
<b>2 ZÁKLADNÍ POJMY A VZTAHY</b> .....	<b>13</b>
2.1 OKRUHY A TĚLESA .....	13
2.2 DIOFANTICKÉ ROVNICE.....	14
2.2.1 Maticové diofantické rovnice.....	15
2.3 PŘENOSOVÁ FUNKCE .....	15
2.4 POPIS MNOHOROZMĚRNÝCH SYSTÉMŮ.....	17
2.4.1 Polynomiální maticové zlomky.....	18
2.5 ZÁKLADNÍ VLASTNOSTI POLYNOMIÁLNÍCH MATIC .....	19
<b>3 POLYNOMIÁLNÍ METODA NÁVRHU ŘÍZENÍ MNOHOROZMĚRNÝCH SYSTÉMŮ</b> .....	<b>20</b>
<b>II PRAKTICKÁ ČÁST</b> .....	<b>23</b>
<b>4 POUŽITÉ VÝVOJOVÉ PROSTŘEDKY</b> .....	<b>24</b>
4.1 OBJEKTIVĚ ORIENTOVANÉ PROGRAMOVÁNÍ .....	24
4.2 VÝVOJOVÉ PROSTŘEDÍ VISUAL STUDIO A PROGRAMOVACÍ JAZYK C#.....	24
4.2.1 .NET Framework.....	25
4.2.2 Programovací jazyk C# .....	25
4.2.3 Microsoft Visual Studio .....	27
<b>5 ALGORITMUS ŘEŠENÍ</b> .....	<b>29</b>
<b>6 VÝSLEDNÁ APLIKACE</b> .....	<b>35</b>
6.1 TŘÍDA POLYNOM.....	35
6.2 TŘÍDA MATRIX.....	35
6.3 TŘÍDA REDUCTION .....	37
6.4 UŽIVATELSKÉ ROZHRANÍ.....	37
6.4.1 Zadání vstupních dat a výpočet.....	38
6.4.2 Možnosti exportu .....	41
6.4.3 Okno nastavení.....	42
<b>7 UKÁZKOVÉ PŘÍKLADY</b> .....	<b>44</b>

---

7.1	ÚLOHA 1 .....	44
7.2	ÚLOHA 2 .....	47
7.3	ÚLOHA 3 .....	50
7.4	ÚLOHA 4 .....	53
<b>ZÁVĚR .....</b>		<b>57</b>
<b>ZÁVĚR V ANGLIČTINĚ.....</b>		<b>58</b>
<b>SEZNAM POUŽITÉ LITERATURY.....</b>		<b>59</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>		<b>60</b>
<b>SEZNAM OBRÁZKŮ .....</b>		<b>61</b>
<b>SEZNAM PŘÍLOH.....</b>		<b>62</b>

## ÚVOD

V dnešní době stále roste význam automatického řízení a s tím i rozvoj teorií a přístupů v tomto oboru. Řízené systémy jsou stále komplexnější. Často je třeba navrhovat řízení pro systémy s více vstupními a výstupními veličinami, které jsou součástí dnešních sofistikovaných technologických procesů. Veličiny těchto mnohorozměrných systémů jsou spolu provázány a vzájemně se ovlivňují.

Návrh řízení mnohorozměrných systémů polynomiálním přístupem lze zařadit do širšího celku - algebraické teorie automatického řízení. Jde o moderní teorii v oboru automatického řízení. Jako v "*klasické*" teorii se vychází ze vstupně - výstupního popisu. Nicméně užitím algebraických metod je možné, oproti metodám "*klasickým*", nalézt zákony řízení pro širší třídu řízených procesů a různé typy signálů a také respektovat celou řadu požadavků na kvalitu řízení. Tyto metody se dnes uplatňují např. v oblasti adaptivního řízení. [1]

V rámci této práce je vytvořena počítačová podpora návrhu řízení mnohorozměrných systémů polynomiálním přístupem. Práce je rozdělena do dvou částí - teoretické a praktické. V teoretické části jsou popsány možnosti realizace programové podpory návrhu MIMO řízení polynomiálním přístupem. Jsou zavedeny základní pojmy a vztahy používané v této diplomové práci. Praktická část je věnována popisu použitých vývojových prostředků se zaměřením na objektově orientovaný programovací jazyk C# a platformu .NET. Následuje kapitola věnovaná použitému algoritmu řešení. Ta mimo jiné obsahuje vývojové diagramy tohoto algoritmu. Nakonec je prezentována funkčnost vytvořené aplikace na čtyřech ukázkových úlohách.

## **I. TEORETICKÁ ČÁST**

# 1 SOUČASNÝ STAV PROGRAMOVÉ PODPORY NÁVRHU MIMO ŘÍZENÍ

Co se programové podpory návrhu řízení (obecně) týče, je dnes patrně nejlepším nástrojem MATLAB (prostředí pro vědeckotechnické výpočty), který v sobě kombinuje jak nástroje pro návrh řízení tak i simulaci výsledného regulačního obvodu. Alternativou může být prostředí Mathematica od firmy Wolfram Research.

Co se týká konkrétně podpory návrhu MIMO řízení polynomiálním přístupem, je k dispozici Polynomial Toolbox pro MATLAB. Tento toolbox nabízí mimo jiné zejména podporu řešení maticových diofantických rovnic. Řešení těchto rovnic představuje stěžejní úlohu v procesu návrhu MIMO řízení polynomiálním přístupem. [8,9]

## 1.1 MATLAB

MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, analýzu dat, paralelní výpočty, měření a zpracování signálů, návrh a simulaci řídicích a komunikačních systémů. Systém MATLAB představuje standard v oblasti technických výpočtů a simulací pro vědu, výzkum, průmysl i vzdělávání. MATLAB poskytuje rozsáhlé specializované knihovny funkcí, které jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti. Nejpodstatnější součástí numerického jádra MATLABu jsou algoritmy pro operace s maticemi reálných a komplexních čísel. MATLAB umožňuje také zobrazení a prezentaci výsledků získaných výpočtem pomocí různých grafů. Základním nástrojem výpočetního systému je uživatelské rozhraní MATLAB Desktop. Pracovní nástroje jako prohlížeč adresářů a souborů, prohlížeč pracovního prostoru, okno historie příkazů, interaktivní spouštěč aplikací, editor, debugger, profiler, hypertextová nápověda a příkazové okno jsou do prostředí plně integrovány. MATLAB je koncipován tak, aby umožňoval i programování aplikací. Programovací jazyk obsahuje všechny nezbytné příkazy pro psaní programů, jako podmíněné příkazy, příkazy pro větvení, cykly atd. Hlavní oblasti využití jsou:

- Aplikovaná matematika
- Fyzika
- Automatické řízení a regulace

- Zpracování signálů
- Zpracování obrazu
- Finanční analýza

[8]

### 1.1.1 Polynomial Toolbox

Otevřená architektura MATLABu vedla ke vzniku knihoven funkcí, nazývaných toolboxy, které rozšiřují použití programu v příslušných oborech. Tyto knihovny, navržené v jazyce MATLABu, nabízejí specializované funkce, které je možno rozšiřovat, modifikovat.

Polynomial Toolbox je balík funkcí a nástrojů pro analýzu i syntézu systémů založený na pokročilých polynomiálních metodách. Skládá se z přibližně 200 M-souborů (soubor se zdrojovým kódem MATLABu). Polynomial Toolbox mimo jiné podporuje práci s polynomiálními maticemi a poskytuje funkce pro řešení maticových diofantických rovnic (funkce `axbyc`). [8,9]

Polynomial toolbox je mocný nástroj, ale na naší fakultě není běžně dostupný. Pro realizaci aplikace byl zvolen programovací jazyk C# a platforma .NET. Ke spuštění aplikace tak není nutné vlastnit MATLAB a uživatel se vyhne problémům s rozdílnými verzemi prostředí MATLAB. MATLAB je využíván pouze pro účely simulace. Výsledná aplikace generuje .m soubory pro MATLAB, které obsahují podklady pro simulační ověření navrženého regulátoru.

## 2 ZÁKLADNÍ POJMY A VZTAHY

Obecně je cílem řízení soustavy  $S$ , získat výstup  $y$ , který splňuje dané požadavky - nejčastěji sledování žádané (řídící, referenční) veličiny  $w$ . Toho je možno dosáhnout vhodnými změnami vstupu soustavy  $u$ . Tyto změny realizuje regulátor  $R$ . Hlavním požadavkem je minimalizace (ideálně odstranění) regulační odchylky  $e = w - y$  v konečném čase. Dále také např. stabilita regulačního obvodu a kompenzace poruch. [4]

### 2.1 Okruhy a tělesa

Okruhem nazýváme algebraickou strukturu definovanou na neprázdné množině  $M$  dvěma binárními operacemi, a to sčítáním  $(+)$  a násobením  $(\cdot)$ . Operace  $(\cdot)$  je distributivní vzhledem k operaci  $(+)$ .

Pro okruh  $\Omega$  a jeho prvky  $a, b, c \in \Omega$  platí následující axiomy:

- $a + b \in \Omega$
- $a + b = b + a$
- $\exists \Theta \in \Omega \quad a + \Theta = \Theta + a = a$
- $\forall a \in \Omega \quad \exists (-a) \in \Omega \quad a + (-a) = \Theta$
- $a \cdot b \in \Omega$
- $a \cdot b = b \cdot a$
- $\exists e \in \Omega \quad e \cdot a = a \cdot e = a$
- $(a + b) \cdot c = a \cdot c + b \cdot c$  (distributivní zákon)
- $(a + b) + c = a + (b + c)$  a  $(a \cdot b) \cdot c = a \cdot (b \cdot c)$  (asociativní zákon)

Tělesem nazýváme asociativní okruh s jednotkou, jehož každý nenulový prvek je invertovatelný. Tedy platí navíc ještě:

- $\forall a \neq \Theta \in \Omega \quad \exists (a^{-1}) \quad a \cdot a^{-1} = e$  (axiom dělení)

Okruh je tedy množina prvků, ve které lze sčítat (odčítat) a násobit prostřednictvím asociativních, komutativních a distributivních operací. Lze-li v této množině navíc i dělit,

již se jedná o těleso. Příkladem okruhu je množina celých čísel nebo množina polynomů. Příkladem tělesa pak např. množina čísel reálných.

Důležitým pojmem je dělitelnost v okruhu. Platí:

- $a$  dělí  $b \Leftrightarrow \exists d \in \Omega \quad b = a \cdot d$
- $d$  je dělitelem  $a$  i  $b \Leftrightarrow a = a_0 \cdot d \quad b = b_0 \cdot d$
- $d$  je NSD( $a, b$ )  $\Leftrightarrow d$  je dělitelné každým dělitelem  $a, b$

V okruhu polynomů je dělitelnost řešena přes kořeny. Budeme-li uvažovat okruh celých čísel, pak zde můžeme o dělitelnosti rozhodnout rozkladem na prvočinitele. Úlohu lze řešit také pomocí zobecněného Euklidova algoritmu. [2, 5]

## 2.2 Diofantické rovnice

Rovnice ve tvaru:

$$ax + by = c \tag{1}$$

se nazývá diofantická. Je definována v množině, která se nazývá okruhem. Tato rovnice má řešení právě tehdy, když největší společný dělitel  $a$  a  $b$  dělí  $c$ . Diofantická rovnice má nekonečně mnoho řešení, která jsou dána:

$$\begin{aligned} x &= x_0 + bt \\ y &= y_0 - at, \end{aligned} \tag{2}$$

kde  $x_0, y_0$  tvoří partikulární řešení a  $t$  je libovolný prvek daného okruhu.

Pro rovnici (1) v okruhu polynomů (s předpokladem nesoudělných polynomů  $a$  a  $b$ ), při splnění podmínek stupňů polynomů, lze nalézt partikulární řešení např. pomocí metody porovnání neurčitých koeficientů. [2]

### 2.2.1 Maticové diofantické rovnice

Maticová diofantická rovnice může být obecně ve třech tvarech:

- pravá:  $AX + BY = C$  (řešení sloupcovými úpravami)
- bilaterální:  $AX + YB = C$  (nelze řešit)
- levá:  $XA + YB = C$  (řešení řádkovými úpravami)

Diofantická rovnice v bilaterálním tvaru je neřešitelná. Zbylé dvě je možno řešit roznásobením člen po členu, čímž jsou převedeny na skalární polynomiální rovnice nebo výhodněji, pomocí elementárních sloupcových či řádkových úprav. Přičemž elementární sloupcové (resp. řádkové) úpravy jsou:

- výměna sloupců (resp. řádků)
- vynásobení všech prvků ve sloupci (resp. řádku) konstantou
- přičtení polynomiálního násobku sloupce (resp. řádku) k dalšímu sloupci (resp. řádku)

Postup je znázorněn na následujícím schématu:

$$\left( \begin{array}{c|c} A & B \\ I & 0 \\ 0 & I \end{array} \right) \xrightarrow{\text{elementární sloupcové úpravy}} \left( \begin{array}{c|c} C & 0 \\ X & Z_1 \\ Y & Z_2 \end{array} \right) \quad (3)$$

$$\left( \begin{array}{ccc} A & I & 0 \\ B & 0 & I \end{array} \right) \xrightarrow{\text{elementární řádkové úpravy}} \left( \begin{array}{ccc} C & X & Y \\ 0 & Z_1 & Z_2 \end{array} \right),$$

kde je  $\left. \begin{array}{l} Z_1 = -B_o \\ Z_2 = A_o \end{array} \right\}$  otočený maticový zlomek vzhledem k  $A$  respektive  $B$ . [2]

### 2.3 Přenosová funkce

Analýza a syntéza lineárních spojitých dynamických systémů (LSDS) je nejstarší a nejhluběji studovaná disciplína teorie řízení. Je třeba zkoumat, jak působí jedna část systému na jinou, na okolí atd. Vhodným způsobem zjišťování těchto vlivů je určení matematického popisu reálného systému (vytvoření matematického modelu). LSDS jsou popisovány lineárními diferenciálními rovnicemi (systém může být případně popsán jen

v diskretních časových okamžicích - diskretní lineární systémy popsané diferenčními rovnicemi). Při analýze (popis statických a dynamických vlastností systémů, vyšetřování stability, atd.) i syntéze (stanovení struktury a parametrů regulačního obvodu tak, aby byly splněny požadavky na regulační pochod) LSDS je třeba řešit složité diferenciální rovnice či jejich soustavy. Zjednodušení těchto výpočtů přináší Laplaceova transformace. Laplaceova transformace přiřazuje po částech spojitě reálné funkci  $f(t)$  reálné proměnné  $t$  (originál) komplexní funkci  $F(s)$  komplexní proměnné  $s$  podle vztahu:

$$F(s) = L\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt \quad (4)$$

Pro funkci  $f(t)$  musí platit:

- $f(t)$  je alespoň po částech spojitá funkce
- $f(t) = \begin{cases} f(t) & t \geq 0 \\ 0 & t < 0 \end{cases}$
- $f(t) \leq Me^{-\alpha t}$  pro  $M > 0, \alpha > 0$

K popisu chování LSDS lze přistoupit různě. Jsou-li k popisu použity pouze vstupní a výstupní funkce, jde o vnější popis systému. Pokud jsou použity další veličiny (nemusí být přímo měřitelné), jde o vnitřní popis. Vnější popis LSDS je možno vyjádřit několika ekvivalentními způsoby. Jedním z nich je přenosová funkce (dále přenos) systému.

Přenos  $G(s)$  je nejčastějším způsobem popisu LSDS. Je definován jako poměr Laplaceových obrazů výstupní  $Y(s)$  a vstupní veličiny  $U(s)$  při nulových počátečních podmínkách.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}, \quad (5)$$

kde  $n$  je řád přenosu a  $n - m$  je relativní řád.

Čitatele i jmenovatele přenosu je možno rozložit na součin kořenových činitelů. Kořeny čitatele se nazývají nuly  $z$ , kořeny jmenovatele póly  $p$ . Jejich rozložení stanovuje vlastnosti daného systému.

$$G(s) = \frac{b_m (s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)} \quad (6)$$

Přenos je možno vyjádřit také ve tvaru s časovými konstantami:

$$G(s) = \frac{K(\tau_1 s + 1)(\tau_2 s + 1) \cdots (\tau_m s + 1)}{(T_1 s + 1)(T_2 s + 1) \cdots (T_n s + 1)} \quad (7)$$

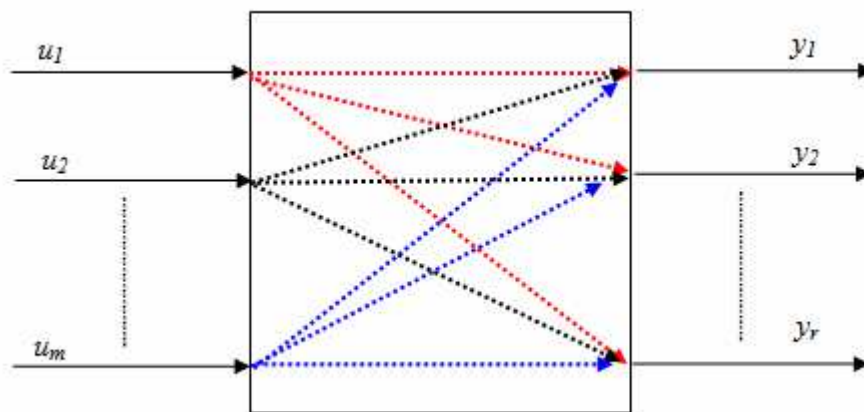
Vztah mezi časovými konstantami, nulami a póly přenosu pro reálné nuly a póly:

$$\tau_i = -\frac{1}{z_i} \quad T_i = -\frac{1}{p_i} \quad (8)$$

[2]

## 2.4 Popis mnohorozměrných systémů

V praxi se často vyskytují systémy s více vstupními a výstupními veličinami. V těchto systémech jsou veličiny provázány vzájemně a obecně může každý vstup ovlivnit kterýkoli výstup – mnohorozměrné systémy. Pro označení mnohorozměrných systémů se používá zkratka „MIMO“ (Multi-Input Multi-Output). Schematické znázornění takového systému s  $m$  vstupy a  $r$  výstupy je na Obr. 1.



Obr. 1. Schéma mnohorozměrného systému

MIMO systém můžeme popsat soustavou diferenciálních rovnic. V případě lineárního systému soustavou lineárních diferenciálních rovnic  $n$ -tého řádu. Vnější popis LDS je možno realizovat přenosovou maticí. Tento popis neobsahuje stavové veličiny, ale popisuje pouze vztahy mezi vstupními a výstupními veličinami systému.

U mnohorozměrných systémů mohou být tyto vztahy přímo odvozeny ze stavového popisu ve tvaru maticových přenosů nebo mohou být použity polynomiální matice a přenosy pak následně vyjádřeny ve tvaru maticových zlomků. Algoritmy řízení použité v této práci jsou založeny právě na vnějším popisu řízeného systému. Matici přenosů můžeme zapsat následovně:

$$G(s) = \begin{pmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1m}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{r1}(s) & G_{r2}(s) & \cdots & G_{rm}(s) \end{pmatrix} = \begin{pmatrix} \frac{g_{11}(s)}{a(s)} & \frac{g_{12}(s)}{a(s)} & \cdots & \frac{g_{1m}(s)}{a(s)} \\ \frac{g_{21}(s)}{a(s)} & \frac{g_{22}(s)}{a(s)} & \cdots & \frac{g_{2m}(s)}{a(s)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{g_{r1}(s)}{a(s)} & \frac{g_{r2}(s)}{a(s)} & \cdots & \frac{g_{rm}(s)}{a(s)} \end{pmatrix} = \frac{G'(s)}{a(s)}, \quad (9)$$

kde  $r$  je počet výstupů a  $m$  počet vstupů systému.

[1,2]

#### 2.4.1 Polynomiální maticové zlomky

Ze vztahů (5) a (9) získáme vektor obrazů výstupu ve tvaru (stále uvažujeme nulové počáteční podmínky):

$$Y(s) = G(s)U(s) = \frac{G'(s)}{a(s)}U(s) \quad (10)$$

Vnější popis systému pomocí přenosové matice (9) nemusí být nejvýhodnější z hlediska odvození zákonů řízení mnohorozměrných systémů. Vhodnějším se jeví vnější popis systému pomocí tzv. polynomiálních matic. Tento popis je vhodnější i z hlediska identifikace těchto systémů.

Přenosovou matici systému (9) je možno za použití (10) zapsat ve tvaru:

$$G(s) = A_L^{-1}(s)B_L(s), \quad (11)$$

kde  $A_L(s) = a(s)I_r$  a  $B_L(s) = G'(s)$ . Reprezentaci přenosové matice ve tvaru (11) nazýváme levý polynomiální maticový zlomek (LPMFD). Existuje také reprezentace přenosové matice ve tvaru nazývaném pravý polynomiální maticový zlomek (RPMFD):

$$G(s) = B_R(s)A_R^{-1}(s), \quad (12)$$

kde  $A_R(s) = a(s)I_m$  a  $B_R(s) = G'(s)$ .

Výhodné je použití reprezentace LPMFD, jelikož je možné získat okamžitě popis systému v časové oblasti (diferenciální rovnice) zpětnou Laplaceovou transformací rovnice

$$Y(s) = A_L^{-1}(s)B_L(s)U(s) \Rightarrow A_L(s)Y(s) = B_L(s)U(s) \quad (13)$$

Naopak, je-li systém popsán diferenciálními rovnicemi, jejich Laplaceovou transformací se získá popis systému v LPMFD. [1,2]

## 2.5 Základní vlastnosti polynomiálních matic

Matice  $P(s)$  o rozměrech  $r \times m$ , jejíž prvky jsou polynomy z okruhu  $\mathfrak{R}[s]$  je polynomiální matice. Množina všech polynomiálních matic rozměru  $r \times m$  je pak  $\mathfrak{R}^{rm}[s]$ .

Tvar polynomiální matice je potom:

$$P(s) = \begin{pmatrix} p_{11}(s) & p_{12}(s) & \cdots & p_{1m}(s) \\ p_{21}(s) & p_{22}(s) & \cdots & p_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ p_{r1}(s) & p_{r2}(s) & \cdots & p_{rm}(s) \end{pmatrix} \quad (14)$$

Pro polynom na pozici  $i, j$  platí:

$$p_{ij}(s) = p_{ij0} + p_{ij1}s + \dots + p_{ijn_{ij}}s^{n_{ij}} \quad (15)$$

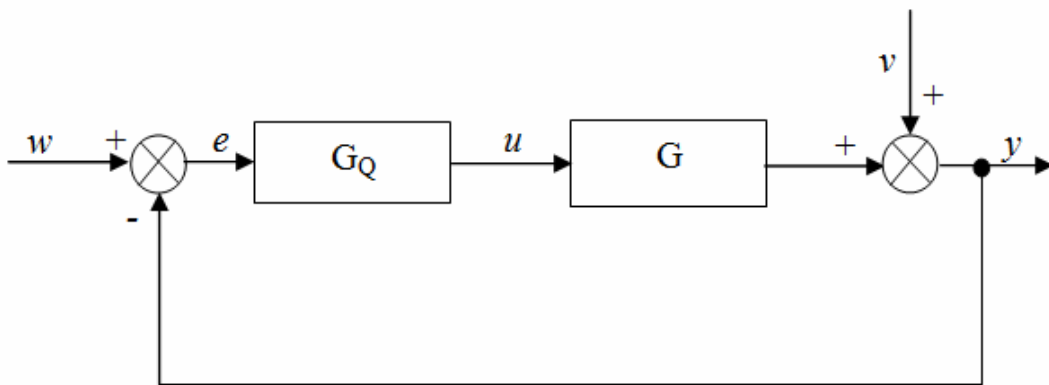
Uvažujme polynomiální matici  $P(s) \in \mathfrak{R}^{rm}[s]$ . Stupeň  $i$ -tého řádku  $r_i(P)$  matice  $P(s)$  je maximálním ze stupňů polynomů v  $i$ -tém řádku. Analogicky stupeň  $j$ -tého sloupce  $c_j(P)$  matice  $P(s)$  je maximálním ze stupňů polynomů v  $j$ -tém sloupci. Stupeň polynomiální matice  $P(s)$ ,  $\deg P(s)$ , je maximální ze stupňů všech polynomů v matici  $P(s)$ .

Všechny v práci uvažované polynomiální matice jsou reálné. To znamená, že koeficienty  $p_{ijk}$ , pro  $i = 1, 2, \dots, r$ ;  $j = 1, 2, \dots, m$ ;  $k = 0, 1, \dots, n_{ij}$  jsou reálné. [1]

### 3 POLYNOMIÁLNÍ METODA NÁVRHU ŘÍZENÍ MNOHOROZMĚRNÝCH SYSTÉMŮ

Vedle klasických metod návrhu regulátorů existují také modernější alternativy vycházející z algebraické teorie. Spočívají v tom, že řešením diofantických rovnic v daném okruhu lze obdržet regulátor, který bude stabilizovat uzavřenou regulační smyčku, a případně splňovat další požadavky. Průběh regulačního pochodu lze ovlivnit volbou charakteristického polynomu uzavřeného regulačního obvodu – metoda umístění pólů. Výhodou polynomiální metody syntézy je, že poskytuje nejen parametry, ale také strukturu regulátoru. Funkčnost metody není omezena na soustavy určitého typu.

Zde bude popisován postup syntézy řízení mnohorozměrných  $t$ -invariantních systémů. Systém řízení v 1DOF konfiguraci je znázorněn na Obr.2.



Obr. 2. Systém řízení

Výstup systém  $G$  je řízen regulátorem  $G_Q$ . Systém má  $r$  vstupních a  $m$  výstupních veličin. Vektor výstupních veličin  $y$  je rozměru  $r \times 1$  a vektor akčních veličin  $u$  je rozměru  $m \times 1$  (výstup regulátoru, vstup řízené soustavy). Vstupem regulačního obvodu je vektor referenčních signálů (žádané hodnoty)  $w$  o rozměrech  $r \times 1$ . Na výstup působí poruchy popsané vektorem  $v$  o rozměru  $r \times 1$ . Řízený systém je popsán ve formě LPMFD (11). Vektory obrazů referenčních signálů a poruch jsou ve tvaru:

$$w(s) = \begin{pmatrix} \frac{h_{w1}(s)}{f_{w1}(s)} & \frac{h_{w2}(s)}{f_{w2}(s)} & \dots & \frac{h_{wr}(s)}{f_{wr}(s)} \end{pmatrix}^T \quad (16)$$

$$v(s) = \begin{pmatrix} \frac{h_{v1}(s)}{f_{v1}(s)} & \frac{h_{v2}(s)}{f_{v2}(s)} & \dots & \frac{h_{vr}(s)}{f_{vr}(s)} \end{pmatrix}^T \quad (17)$$

Přenosová matice odvozeného zpětnovazebního regulátoru bude ve tvaru RPMFD:

$$G_Q(s) = Q_R(s)P_R^{-1}(s), \quad (18)$$

kde  $Q_R(s) \in \mathfrak{R}^{mr}[s]$  a  $P_R(s) \in \mathfrak{R}^{rr}[s]$  jsou zprava nesoudělné polynomiální matice.

Popis základních signálů v regulačním obvodu:

$$u(s) = Q_R(s)D^{-1}(s)A_L(s)[w(s) - v(s)] \quad (19)$$

$$y(s) = [I - P_R(s)D^{-1}(s)A_L(s)]w(s) + P_R(s)D^{-1}(s)A_L(s)v(s) \quad (20)$$

$$e(s) = P_R(s)D^{-1}(s)A_L(s)[w(s) - v(s)], \quad (21)$$

kde  $D(s) = A_L(s)P_R(s) + B_L(s)Q_R(s)$ .

Mnohorozměrný systém řízení, ve kterém přenosová matice řízeného systému je ve tvaru LPMFD je BIBO stabilní (systém je stabilní, pokud omezený vstup do systému generuje omezený výstup ze systému) tehdy, jestliže polynomiální matice v RPMFD stabilizujícího regulátoru jsou dány řešením maticové diofantické rovnice:

$$A_L(s)P_R(s) + B_L(s)Q_R(s) = D(s), \quad (22)$$

kde  $D(s) \in \mathfrak{R}^{rr}[s]$  je stabilní polynomiální matice.

Polynomiální matice  $D(s)$  se volí jako diagonální matice, která obsahuje na své diagonále charakteristické polynomy  $d_i$  (se zvolenými póly). Tvar polynomů  $d_i$  je:

$$(s + m_i)^{\deg d_i}, \quad (23)$$

kde  $m_i$  jsou volené póly. Póly volíme v záporné polorovině komplexní roviny - stabilní charakteristický polynom.

Stupně polynomů ve sloupcích matic  $Q_R(s)$  a  $P_R(s)$  musí splňovat podmínku:

$$c_j(Q_R) \leq c_j(P_R), \quad j = 1, 2, \dots, r \quad (24)$$

[1, 2]

Pro regulační odchylku musí platit:

$$\lim_{t \rightarrow \infty} \|e(t)\| = \lim_{s \rightarrow 0} \|se(s)\| = 0 \quad (25)$$

Aby byla tato podmínka splněna, musí být eliminovány jmenovatele ve vektorech obrazů referenčního signálu (16) i poruchy (17) působících v systému řízení (Obr. 2). Tyto jmenovatele je možno odstranit pouze, pokud platí:

$$P_R(s) = F(s)\tilde{P}_R(s), \quad (26)$$

kde matice  $F(s)$  je diagonální matice s polynomem  $f(s)$  na své diagonále:  $F(s) = f(s)I_r$ . Polynom  $f(s)$  je dělitelný současně všemi jmenovateli prvků vektorů  $w(s)$  a  $v(s)$ . Jinými slovy polynom  $f(s)$  je nejmenší společný násobek těchto jmenovatelů. Rovnice (18) má potom tvar:

$$A_L(s)F(s)\tilde{P}_R(s) + B_L(s)Q_R(s) = D(s) \quad (27)$$

Přenosová matice výsledného zpětnovazebního regulátoru je poté:

$$G_Q(s) = Q_R(s)(F(s)\tilde{P}_R(s))^{-1} \quad (28)$$

[1]

Rovnici (27) je možno výhodně řešit pomocí elementárních sloupcových úprav. Postup je znázorněn na následujícím schématu odvozeném z (3):

$$\left( \begin{array}{c|c} A_L F & B_L \\ \hline I & 0 \\ \hline 0 & I \end{array} \right) \xrightarrow{\text{elementární sloupcové úpravy}} \left( \begin{array}{c|c} D & 0 \\ \hline \tilde{P}_R & -B_R \\ \hline Q_R & A_R \end{array} \right) \quad (29)$$

[2]

Pozn.: V (29) se pracuje v podstatě s jednou polynomiální maticí vzniklou spojením příslušných matic. Pro usnadnění orientace bude ovšem tato dále označována pojmem *struktura*, která má svou levou a pravou část.

## **II. PRAKTICKÁ ČÁST**

## 4 POUŽITÉ VÝVOJOVÉ PROSTŘEDKY

### 4.1 Objektově orientované programování

Objektově orientované programování (OOP) je způsob programování založený na objektech. Využívá programovací jazyk k napodobení objektů skutečného světa. Objekty jsou definovány na základě třídy. Třída je jakousi šablonou - vzorem pro objekt. Objekt je instancí třídy. Třída se skládá z atributů (datové položky) a metod (členských funkcí), které s nimi pracují a definují tak chování objektu.

Třída je šablona, která deklaruje atributy a definuje metody objektů. Třída je popsána definicí třídy. Atributy i metody třídy mají při deklaraci nastavenou jednu ze dvou úrovní viditelnosti - private nebo public. Položky private jsou přístupné pouze v metodách dané třídy.

Hlavní vlastnosti OOP:

- Abstrakce
- Zapouzdřenost - spojení dat a metod v jeden celek. Datové položky by měly být skryté pro jiné objekty a přístupné pouze pomocí metod objektu. Metody jsou rozhraním objektů.
- Dědičnost umožňuje vytvořit novou třídu na základě existující třídy.
- Polymorfismus - objekty různých tříd reagují na stejnou zprávu různě (různé chování). Několik objektů může poskytovat stejné rozhraní, pracuje se s nimi navenek stejně, ale jejich konkrétní chování je různé.

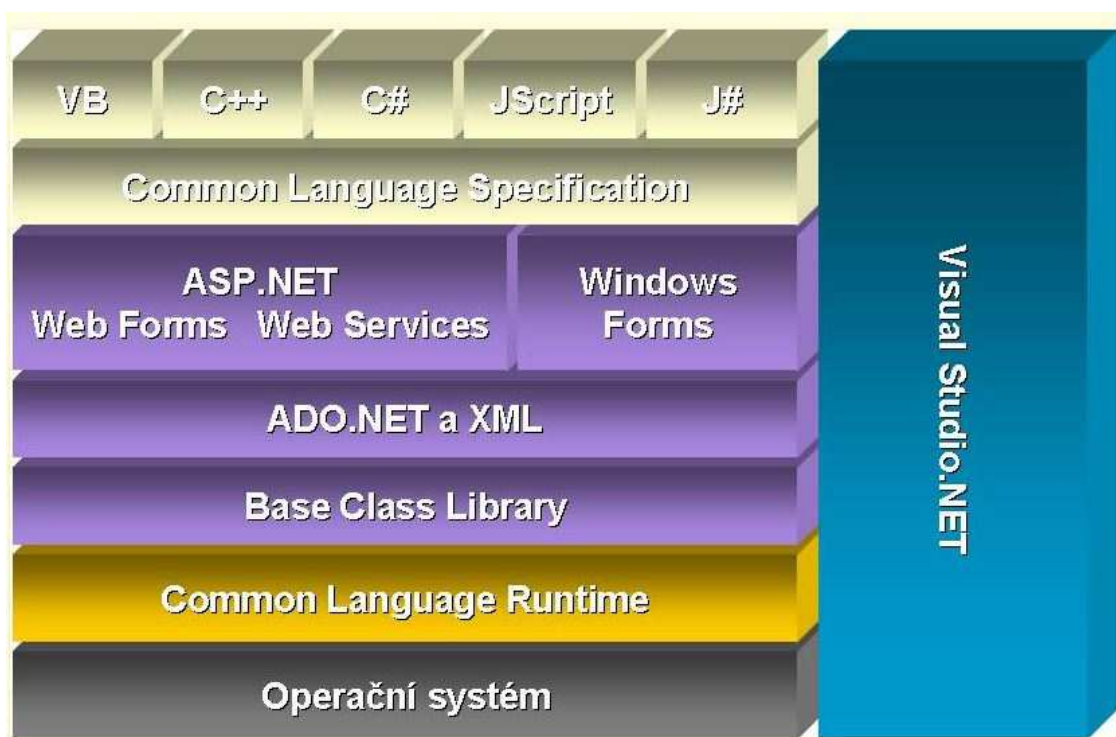
[10]

### 4.2 Vývojové prostředí Visual Studio a programovací jazyk C#

Výsledná aplikace byla naprogramována v programovacím jazyce C# pro platformu .NET Framework 4 pomocí vývojového prostředí Visual Studio 2010 verze 10.0.30319.1 od společnosti Microsoft.

#### 4.2.1 .NET Framework

Platforma pro vývoj aplikací .NET („dotnet“) poskytuje podporu pro tvorbu, nasazení a běh aplikací pod operačním systémem Windows. Prostředí .NET Framework (Obr. 3) je potřebné pro běh aplikací (nabízí např. potřebné knihovny) založených na platformě .NET. Základem je virtuální stroj Common Language Runtime (CLR), který realizuje základní infrastrukturu, nad kterou je framework vybudován. Poskytuje mimo jiné správu paměti. Nad CLR se nachází několik hierarchicky umístěných knihoven, které jsou rozděleny do jmenných prostorů. Mezi těmito knihovnami je i sada knihoven pro tvorbu GUI. Platforma .NET nepředepisuje použití žádného konkrétního programovacího jazyka.



Obr. 3. Architektura .NET Framework

[6,7]

#### 4.2.2 Programovací jazyk C#

Programovací jazyk C# je mnohoúčelový, objektově orientovaný programovací jazyk navržený pro tvorbu široké škály aplikací pod .NET Framework. Vychází z programovacích jazyka C++ (většina syntaxe z jazyka C). Programování v jazyce C# přináší velkou produktivitu. Jazyk sám nepřináší žádné speciální datové typy, nekompatibilní s jinými jazyky. [6]

Ukázka zdrojového kódu v C#. Konkrétně jde o realizaci násobení polynomu konstantou ve výsledné aplikaci (přetěžování operátorů):

```
using System;

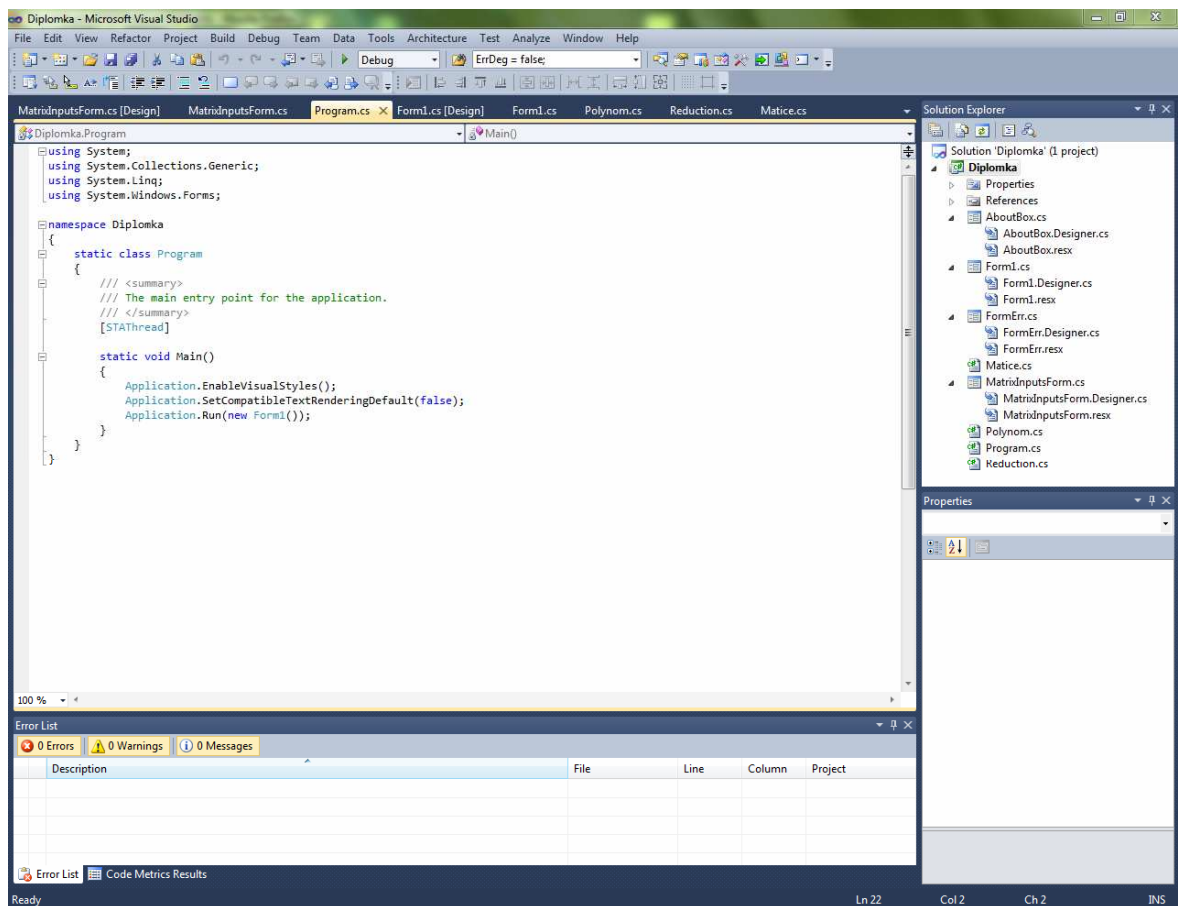
namespace Diplomka
{
    public class Polynom /*trida reprezentujici polynom*/
    {
        /* ... */

        public static Polynom operator *(double k, Polynom p_a)
        {
            int delka = p_a.GetLength();
            double[] vys = new double[delka];
            for (int i = 0; i < delka; i++)
            {
                vys[i] = k * p_a.polynom[i];
            }
            Polynom vysledek = new Polynom();
            vysledek.polynom = vys;
            return vysledek;
        }
        /* ... */
    }
}
```

### 4.2.3 Microsoft Visual Studio

Microsoft Visual Studio (Obr. 4) je oficiální vývojové prostředí od firmy Microsoft. Visual Studio obsahuje mimo jiné:

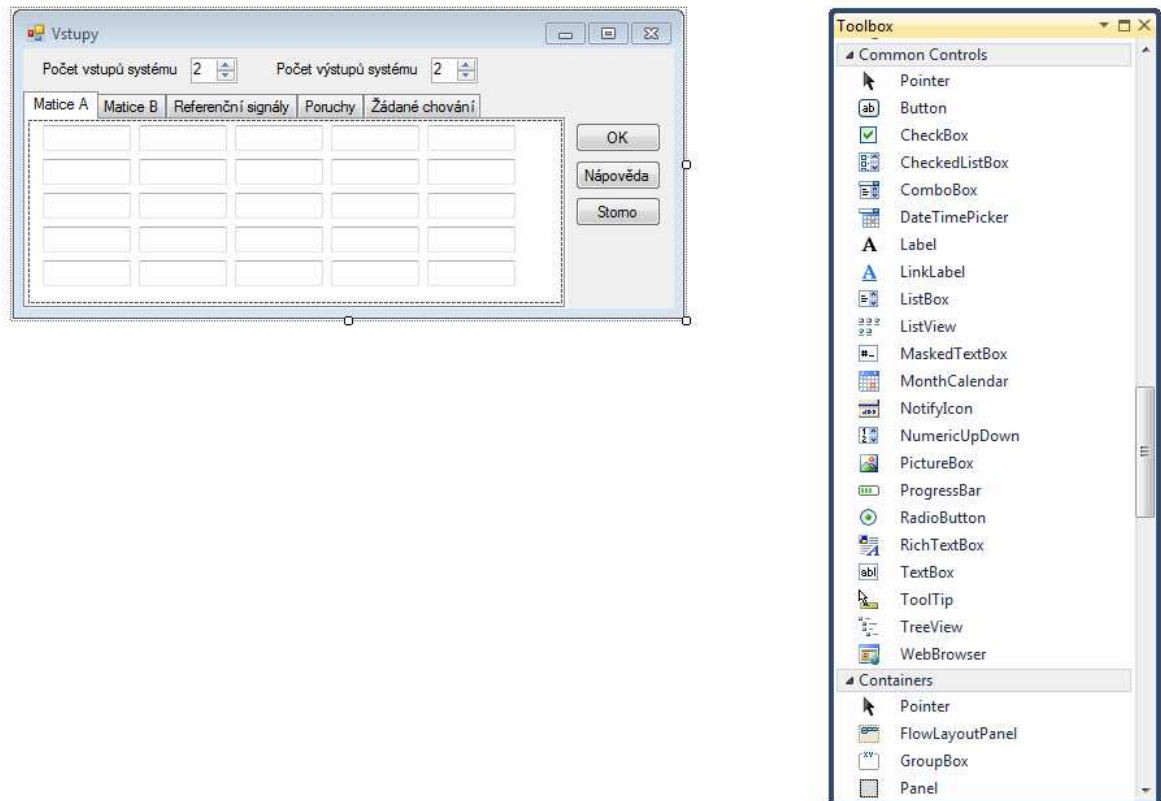
- editor vlastního zdrojového kódu
- debugger
- nástroj pro design GUI formulářů
- nástroj pro design webových stránek
- nástroj pro tvorbu tříd



Obr. 4. Microsoft Visual Studio

Mezi hlavní podporované programovací jazyky patří C# a Visual Basic. Pomocí rozšíření však může podporovat jakýkoliv programovací jazyk.

Editor kódu nabízí zvýraznění syntaxe i chyb v kódu, automatické dokončování, vkládání záložek do kódu, vyhledávání atd. S vývojem GUI aplikací pomáhá nástroj pro design GUI formulářů fungující na principu drag and drop (Obr. 5).



Obr. 5. Nástroj pro design GUI formulářů

## 5 ALGORITMUS ŘEŠENÍ

Pro návrh algoritmu bylo výchozím schéma řešení úlohy (29). Vstupem algoritmu jsou příslušné polynomiální matice  $(A_L, F, B_L, D)$ . Výstupem jsou polynomiální matice  $P_R$  a  $Q_R$ , které představují navrhovaný regulátor ve tvaru (28).

Ukázkový příklad:

Uvažujme MIMO systém se dvěma vstupy a dvěma výstupy. Tento systém je popsán diferenciálními rovnicemi ve tvaru:

$$\begin{aligned} y_1'(t) + y_1(t) &= u_1(t) + 2u_2(t) \\ 2y_1(t) + y_2'(t) + 3y_2(t) &= 2u_1(t) + 3u_2(t) \end{aligned}$$

Obraz těchto rovnic po Laplaceově transformaci (4):

$$\begin{aligned} sY_1(s) + Y_1(s) &= U_1(s) + 2U_2(s) \\ 2Y_1(s) + sY_2(s) + 3Y_2(s) &= 2U_1(s) + 3U_2(s) \end{aligned}$$

Systém je možno zapsat ve tvaru LPMFD (10,11):

$$\begin{pmatrix} s+1 & 0 \\ 2 & s+3 \end{pmatrix} \begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix}$$

$$A_L = \begin{pmatrix} s+1 & 0 \\ 2 & s+3 \end{pmatrix} \quad B_L = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

Úkolem je navrhnout spojitý regulátor pro tento systém. Budeme předpokládat žádanou hodnotu a poruchu ve tvaru skoku:

$$w(t) = (h_{w1} \cdot 1(t) \quad h_{w2} \cdot 1(t))^T \quad v(t) = (h_{v1} \cdot 1(t) \quad h_{v2} \cdot 1(t))^T$$

$$w(s) = \begin{pmatrix} \frac{h_{w1}}{s} & \frac{h_{w2}}{s} \end{pmatrix}^T \quad v(s) = \begin{pmatrix} \frac{h_{v1}}{s} & \frac{h_{v2}}{s} \end{pmatrix}^T$$

Tedy matice  $F$  bude ve tvaru:

$$F = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix}$$

Zvolíme dvojnásobné póly  $p_1 = -1$  a  $p_2 = -2$ . Matice  $D$  tedy bude:

$$D = \begin{pmatrix} (s+1)^2 & 0 \\ 0 & (s+2)^2 \end{pmatrix}$$

Je třeba řešit maticovou diofantickou rovnici  $A_L F \tilde{P}_R + B_L Q_R = D$ . využijeme schéma (29).

Sloupce označíme pro přehlednost římskými číslicemi.

$$\left( \begin{array}{c|c} A_L F & B_L \\ \hline I & 0 \\ 0 & I \end{array} \right) \xrightarrow{\text{elementární sloupcové úpravy}} \left( \begin{array}{c|c} D & 0 \\ \hline \tilde{P}_R & -B_R \\ Q_R & A_R \end{array} \right)$$

$$\left( \begin{array}{cc|cc} s^2+s & 0 & 1 & 2 \\ 2s & s^2+3s & 2 & 3 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right) \xrightarrow[\text{IV}=-1\cdot\text{IV}]{\text{IV}=-2\cdot\text{III}+\text{IV}} \left( \begin{array}{ccc|c} s^2+s & 0 & 1 & 0 \\ 2s & s^2+3s & 2 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & -1 \end{array} \right) \xrightarrow{\text{III}=-2\cdot\text{IV}+\text{III}}$$

$$\left( \begin{array}{cccc} s^2+s & 0 & 1 & 0 \\ 2s & s^2+3s & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -3 & 2 \\ 0 & 0 & 2 & -1 \end{array} \right) \xrightarrow{\text{I}=(s+1)\cdot\text{III}+\text{I}} \left( \begin{array}{cccc} s^2+2s+1 & 0 & 1 & 0 \\ 2s & s^2+3s & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3s-3 & 0 & -3 & 2 \\ 2s+2 & 0 & 2 & -1 \end{array} \right) \xrightarrow{\text{II}=(s+4)\cdot\text{IV}+\text{II}}$$

$$\left( \begin{array}{cccc} s^2+2s+1 & 0 & 1 & 0 \\ 2s & s^2+4s+4 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3s-3 & 2s+8 & -3 & 2 \\ 2s+2 & -s-4 & 2 & -1 \end{array} \right) \xrightarrow{\text{I}=(-2s)\cdot\text{IV}+\text{I}}$$

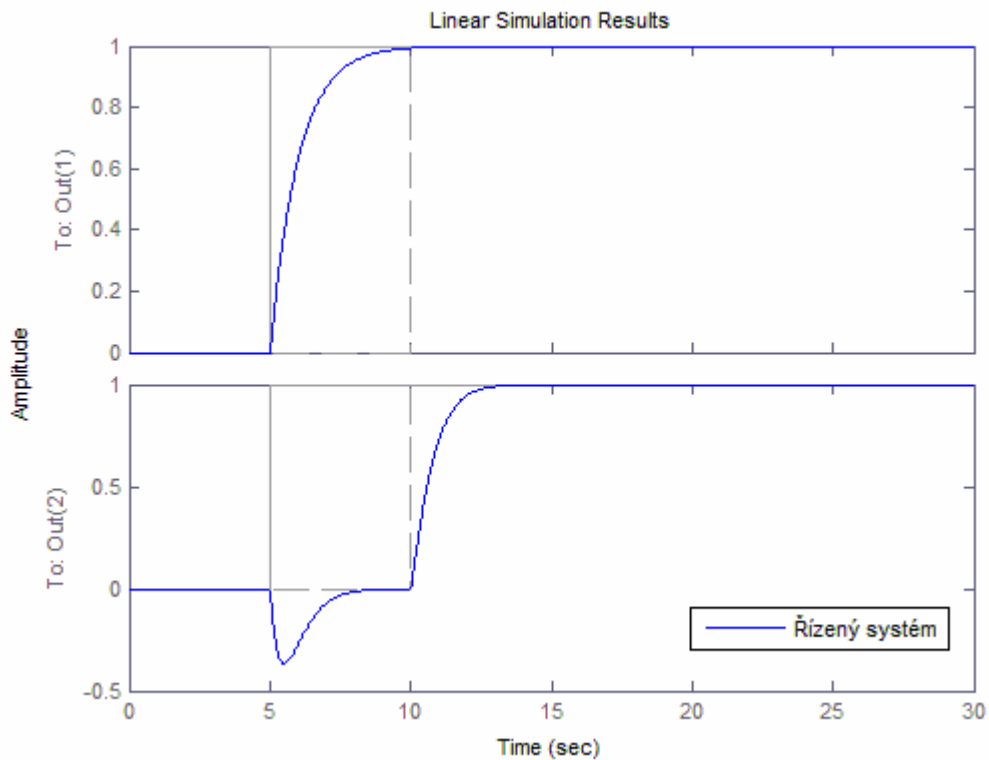
$$\left( \begin{array}{cccc} s^2+2s+1 & 0 & 1 & 0 \\ 0 & s^2+4s+4 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -7s-3 & 2s+8 & -3 & 2 \\ 4s+2 & -s-4 & 2 & -1 \end{array} \right)$$

$$\tilde{P}_R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad Q_R = \begin{pmatrix} -7s-3 & 2s+8 \\ 4s+2 & -s-4 \end{pmatrix}$$

Výsledný regulátor má tedy tvar (28):

$$G_Q(s) = \begin{pmatrix} -7s-3 & 2s+8 \\ 4s+2 & -s-4 \end{pmatrix} \left( \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right)^{-1} = \begin{pmatrix} \frac{-7s-3}{s} & \frac{2s+8}{s} \\ \frac{4s+2}{s} & \frac{-s-4}{s} \end{pmatrix}$$

Simulace regulačního pochodu je na Obr. 6.



Obr. 6. Odezva systému na skokovou změnu žádaných hodnot

Algoritmus řešení výše uvedené úlohy lze rozdělit do 2 hlavních kroků (stále vycházíme ze schématu (29)):

- Vytvoření jednotkové matice na pozici polynomiální matice  $B_L$  (Obr. 7).
- Vytvoření polynomiální matice  $D$  na pozici polynomiální matice  $A_L \cdot F$  (Obr. 8).

Ke správné interpretaci vývojových diagramů na Obr. 7 a 8 je třeba dodat, že zachycují základní myšlenku algoritmu. Ve výsledné aplikaci je tento algoritmus rozšířen o několik ošetření chybových stavů. Ve zdrojovém kódu je navíc i obsluha zápisu postupu řešení atd. Při zahrnutí těchto dodatečných operací by byly vývojové diagramy neúnosně rozsáhlé. Kompletní zdrojový kód realizující tento algoritmus je implementován jako metoda Solve ve třídě Reduction (více viz. kapitola 6.3).

V diagramech je použito následující značení:

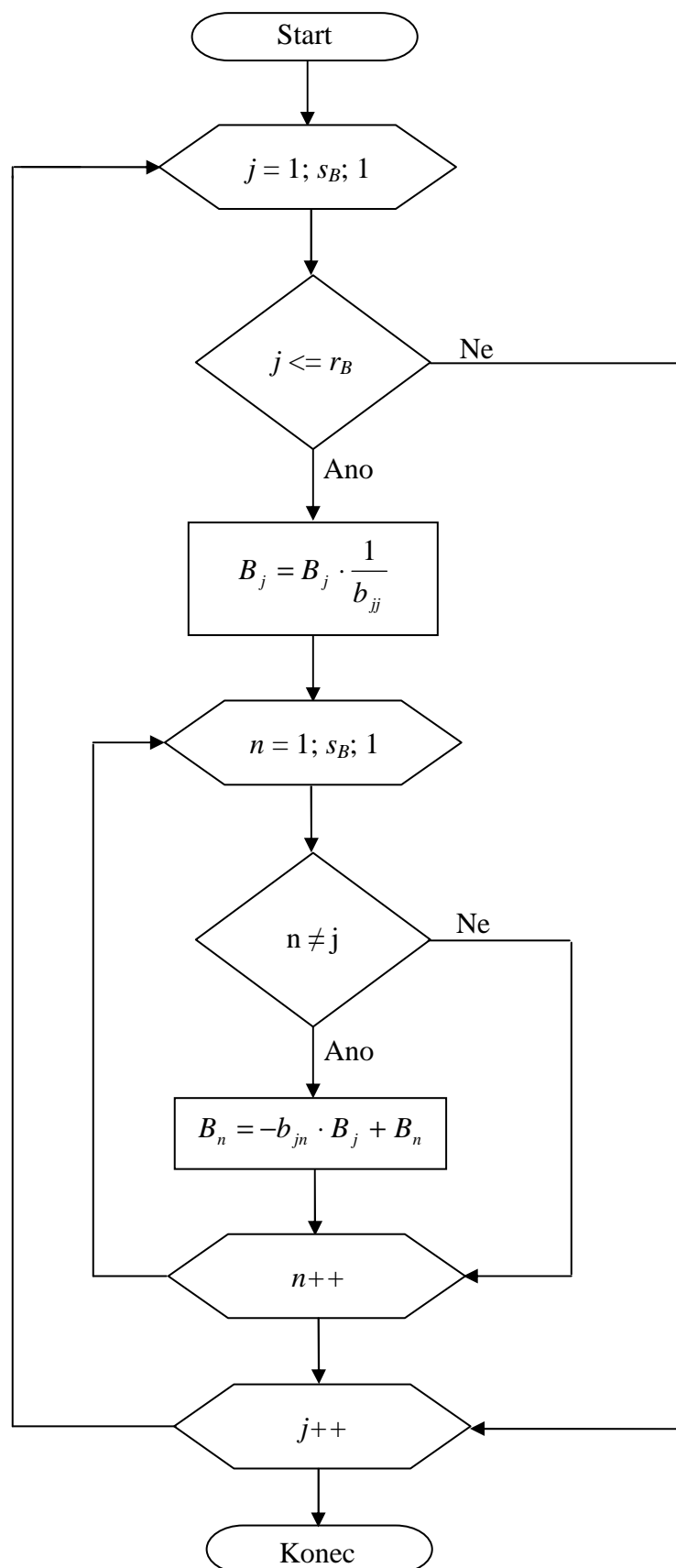
- $A_x$  -  $x$ -tý sloupec v levé části struktury (viz. níže)
- $B_x$  -  $x$ -tý sloupec v pravé části struktury (viz. níže)
- $a_{ij}$  - prvek matice  $A$  na řádku  $i$  ve sloupci  $j$
- $b_{ij}$  - prvek matice  $B$  na řádku  $i$  ve sloupci  $j$
- $d_{ij}$  - prvek matice  $D$  na řádku  $i$  ve sloupci  $j$
- $r_A$  - počet řádků matice  $A$
- $s_A$  - počet sloupců matice  $A$
- $r_B$  - počet řádků matice  $B$
- $s_B$  - počet sloupců matice  $B$
- Platí  $r_A = r_B$ .

Výchozím je následující značení v rámci schématu (29):

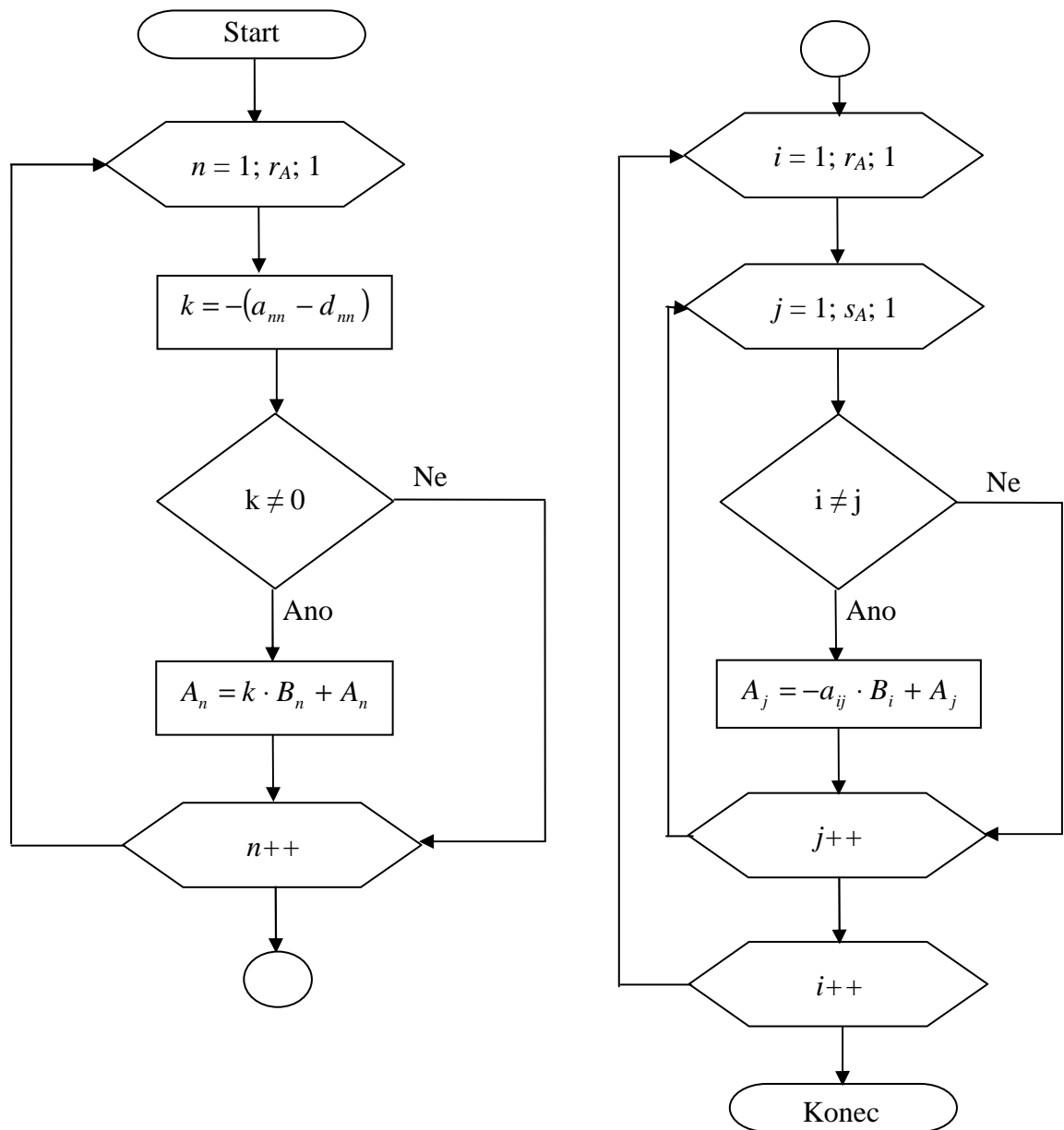
$$\left( \begin{array}{c|ccc} A & B \\ \hline I & 0 \\ 0 & I \end{array} \right) = \left( \begin{array}{ccc|ccc} a_{11} & \cdots & a_{1s_A} & b_{11} & \cdots & b_{1s_B} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{r_A 1} & \cdots & a_{r_A s_A} & b_{r_B 1} & \cdots & b_{r_B s_B} \\ 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{array} \right)$$

Tedy pro upřesnění vzhledem k (29) bylo pro zjednodušení užito substituce  $A = A_L F$

a  $B = B_L$ . Tak např.  $B_1 = (b_{11} \quad \cdots \quad b_{r_B 1} \quad 0 \quad \cdots \quad 0 \quad 1 \quad \cdots \quad 0)^T$



Obr. 7. Vývojový diagram pro tvorbu jednotkové matice



Obr. 8. Polynomiální matice  $D$  na pozici  $AL \cdot F$

## 6 VÝSLEDNÁ APLIKACE

Celá aplikace je naprogramována v programovacím jazyce C#. Zadávání vstupů a zobrazení výstupů je realizováno pomocí GUI. Program se skládá z několika tříd. Mimo tříd jednotlivých GUI formulářů, to jsou:

- Polynom
- Matrix
- Reduction

### 6.1 Třída Polynom

Třída Polynom obsahuje definici polynomu a operací s těmito polynomy. Polynom je reprezentován jako pole prvků double. Třída dále obsahuje několik konstruktorů. Konstruktor bez parametru vytvoří polynom o jednom prvku 0. Dále je k dispozici konstruktor polynomu z textového řetězce, z čísla typu double (vytvoří polynom nultého řádu o hodnotě předaného čísla) a z pole prvků typu double.

- `public Polynom()`
- `public Polynom(String s)`
- `public Polynom(double x)`
- `public Polynom(double[] x)`

Třída dále obsahuje několik metod. Např. metodu pro převod polynomu na datový typ String ve tvaru vhodného pro výstup. Nakonec jsou v rámci třídy realizovány základní operace s polynomy pomocí přetěžování operátorů. Konkrétně jde o sčítání (odčítání) a násobení (jiným polynomelem i číselnou konstantou typu double).

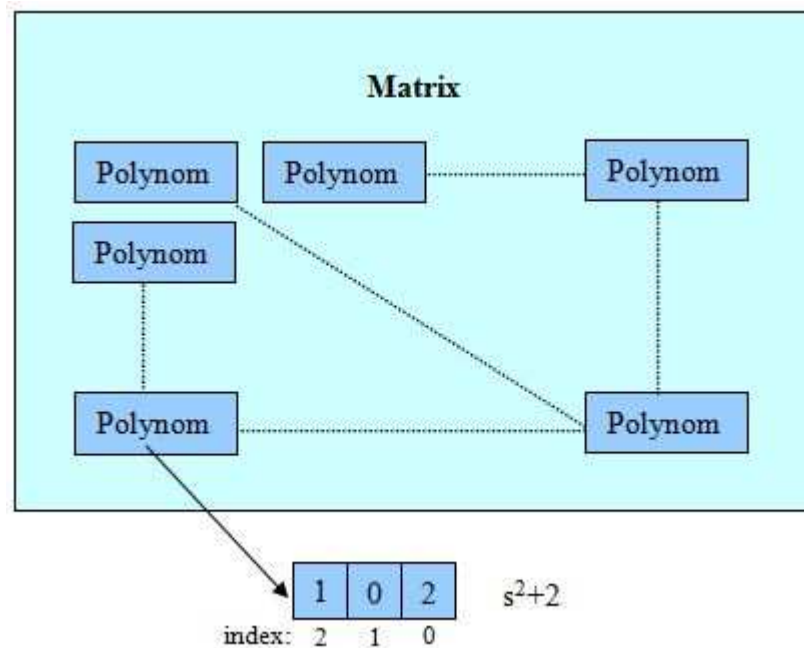
### 6.2 Třída Matrix

Tato třída zahrnuje definici polynomiálních matic a operací s nimi. Polynomiální matice je reprezentována jako dvourozměrné pole objektů třídy Polynom (Obr. 9). Třída opět obsahuje několik konstruktorů. Prázdný konstruktor vytvoří objekt Matrix, který bude reprezentovat matici  $(0)$ . Dále třída obsahuje konstruktor se vstupním parametrem typu String. Ten vytvoří matici na základě vstupního textového řetězce. Formát řetězce

je následující: znak ; odděluje jednotlivé prvky matice (polynomy) a odřádkování ukončuje řádek matice. Toto formátování nemusí uživatel znát. Vstup je realizován prostřednictvím GUI. Následují speciální konstruktory pro tvorbu matice o rozměrech  $r \times m$  a diagonální matice s prvky stejné hodnoty na své diagonále. Prvek této matice je definován jako objekt třídy Polynom a nebo jako číslo typu double.

Konstruktory:

- public Matrix()
- public Matrix(String s)
- public Matrix(int x, int y, double[] a)
- public Matrix(int x, int y, Polynom a)
- public Matrix(int n, double [] a)
- public Matrix(int n, Polynom a)



Obr. 9. Třída Matrix

Třída Matrix dále obsahuje několik metod. Kromě opětovného převodu na textový řetězec stojí za zmínku i metody pro přímý přístup k jednotlivým polynomům matice. Přetěžováním operátorů jsou realizovány operace s celými maticemi jako např. násobení matic.

### 6.3 Třída Reduction

Objekt této třídy obsahuje specifickou strukturu pro řešení úlohy. V podstatě představuje následující uspořádání objektů třídy matice dle (29):

$$\left( \begin{array}{c|c} A_L F & B_L \\ \hline I & 0 \\ 0 & I \end{array} \right)$$

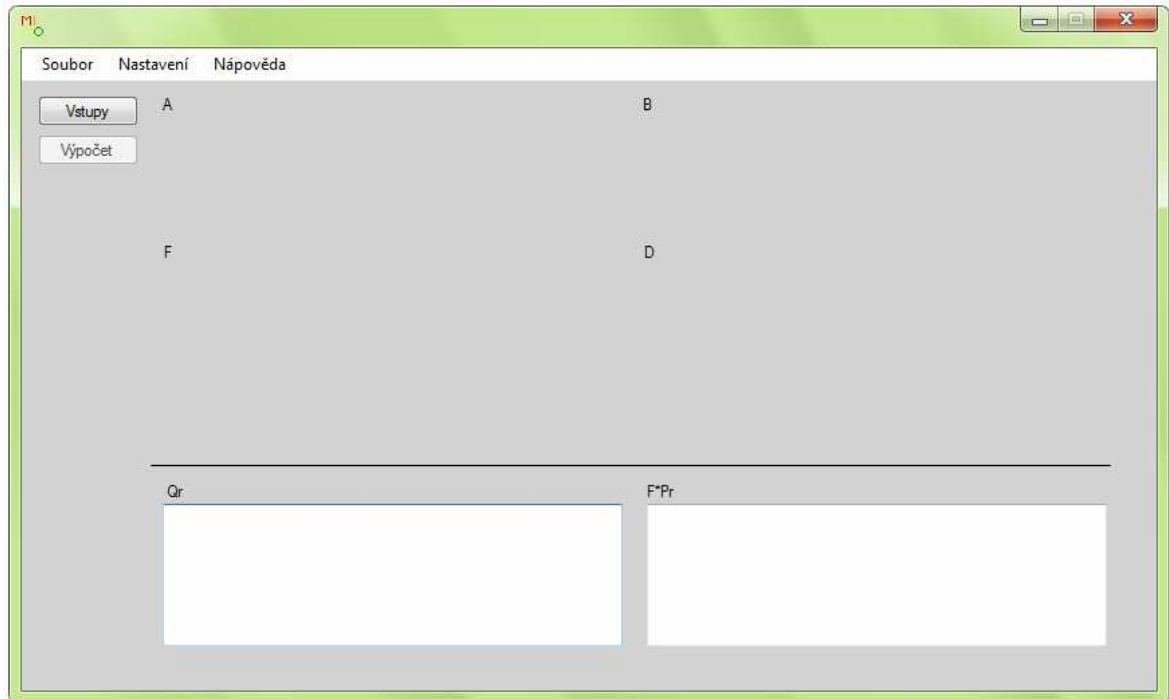
Jde o pole objektů třídy matice. Toto pole má pevně dané rozměry 3x2 a jediný přípustný konstruktor: `public Reduction(Matrix A1, Matrix f, Matrix B1, Matrix mD)`. Tento konstruktor naplní strukturu příslušnými maticemi. Metody této třídy realizují elementární sloupcové úpravy.

- výměna sloupců:  
`public void Changecolumns(int c1, int c2)`
- vynásobení všech prvků ve sloupci konstantou:  
`public void Kcolumn(int col, Polynom k)`
- přičtení polynomiálního násobku sloupce k dalšímu sloupci:  
`public void Multiplication(int c1, int c2, Polynom k)`

Celé řešení dle popsaného algoritmu potom vykoná metoda `Solve()`. Zároveň také vytváří výpis provedených operací, který je posléze možno exportovat do textového souboru.

### 6.4 Uživatelské rozhraní

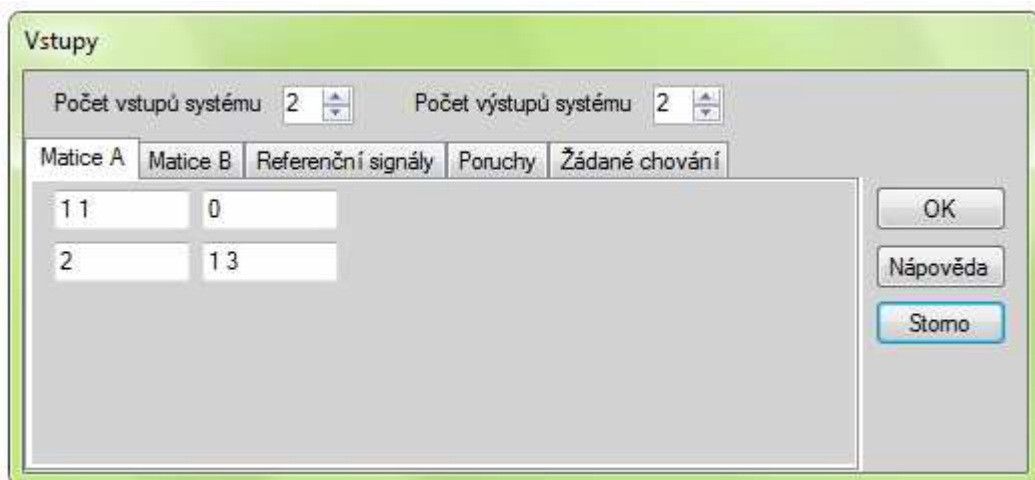
Komunikace mezi uživatelem a aplikací je realizována přes GUI. Po spuštění programu MIMO.exe se zobrazí úvodní obrazovka (Obr. 10). Do programu nejsou zadána žádná vstupní data, a tak je většina procedur zatím nedostupná. Vstupní data (polynomiální matice  $A_L$ ,  $B_L$ ,  $F$ ,  $D$ , tvary žádaných hodnot a poruch) je možno zadat z dialogového okna přístupného přes tlačítko *Vstupy*. Zobrazí se příslušný formulář (Obr. 11).



Obr. 10. Úvodní obrazovka

#### 6.4.1 Zadání vstupních dat a výpočet

Okno pro zadání vstupních dat dovoluje uživateli zadat vstupní data pro výpočet. V okně je již předpřipraveno zadání ukázkové úlohy (viz. kapitola 5). Uživatel si zvolí počet vstupů a výstupů systému. Minimum je systém s jedním vstupem a jedním výstupem. Maximální povolená konfigurace je pak 5 vstupů a 5 výstupů. Na jednotlivých kartách uživatel vyplní požadované parametry.

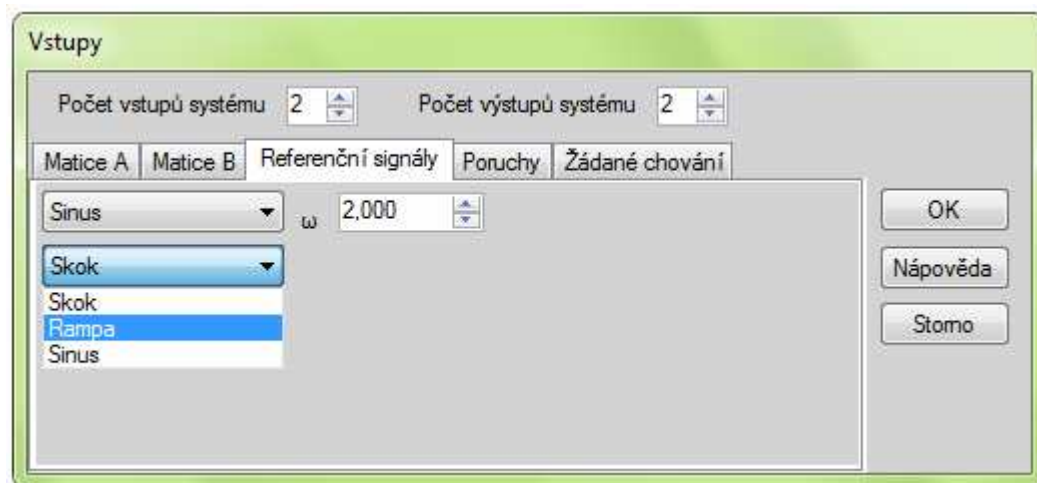


Obr. 11. Okno pro zadání vstupních dat

Na kartě *Matice A* (analogicky na kartě *Matice B*) uživatel zadává matici  $A_L$  respektive  $B_L$ , které popisují řízený systém. Jednotlivé prvky matice (polynomy) se zapisují do připravených polí. Polynomy se zadávají zápisem jednotlivých koeficientů řazených sestupně podle mocniny  $s$ . Tak např. polynom  $s^3 - 3s + 2$  bude uživatelem vložen ve tvaru 1 0 -3 2. Ve vstupních polích jsou jakékoliv neplatné znaky ignorovány. Maximální přípustný stupeň polynomu je 3.

Na dalších dvou kartách je možno zadat jeden ze tří možných požadovaných tvarů referenčních signálů a poruch. Tyto karty nabízí zcela stejné rozhraní (Obr. 12). Je možno zvolit jeden ze 3 tvarů žádané hodnoty:

- skok
- rampu
- harmonický průběh (k dispozici navíc volba hodnoty úhlové frekvence  $\omega$ , kde  $\omega$  je reálné číslo  $\omega > 0$ )

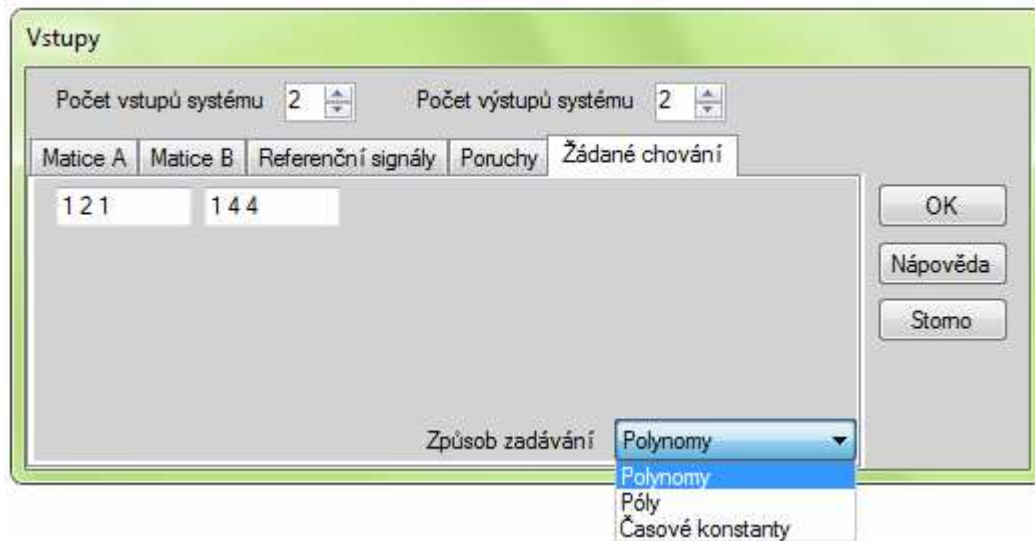


Obr. 12. Zadávání tvaru referenčních signálů

Na poslední kartě (Obr. 13) je možno zadat požadované chování. Na základě tohoto zadání bude vygenerována polynomiální matice  $D(s)$  (diagonální matice; viz. kapitola 3). Je možno zvolit jeden ze tří způsobů zadávání:

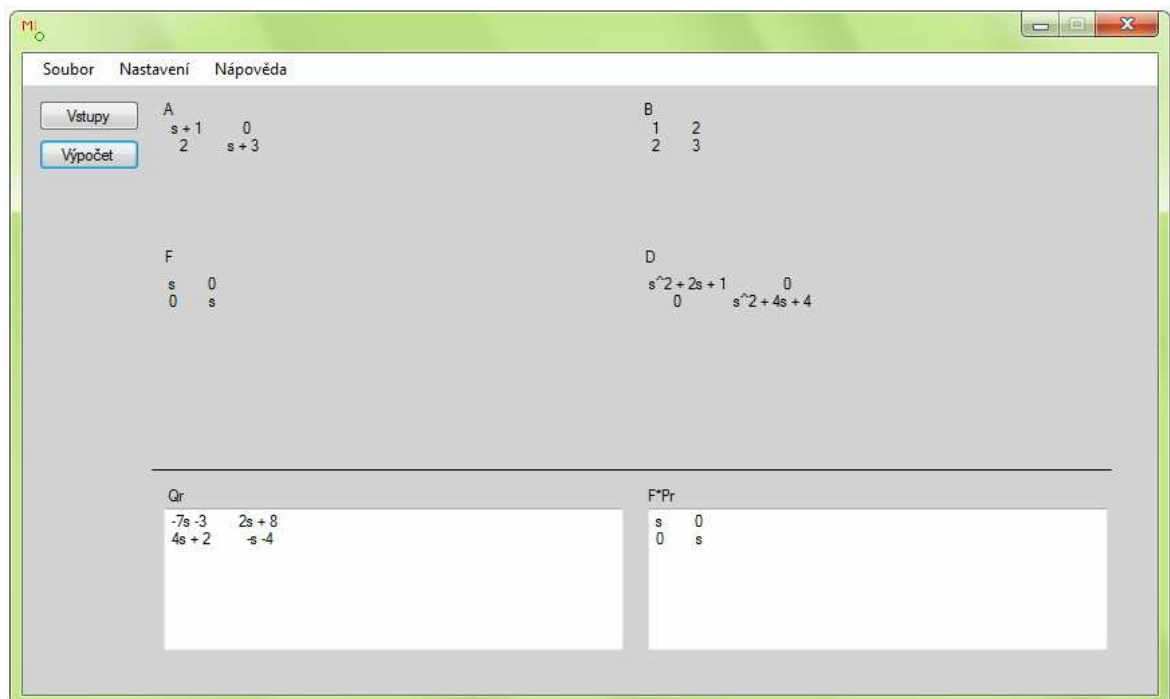
- zadat požadované póly přímo
- zadat požadované chování přímo ve tvaru polynomu

- zadat požadované časové konstanty (vztah mezi časovými konstantami a póly přenosu viz. (8))



Obr. 13. Volba prvků matice  $D(s)$

Volba se potvrzuje stiskem tlačítka *OK*. Matice jsou předány programu k dalšímu zpracování. V případě neplatného zadání (například polynom příliš vysokého stupně) bude uživatel na toto upozorněn a požádán o nápravu. Na hlavní obrazovce (Obr. 14) je poté možno provést syntézu regulátoru po zadaný systém (tlačítko *Výpočet*).



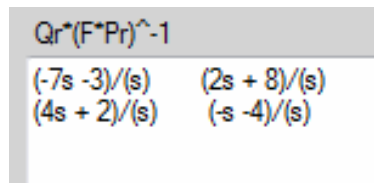
Obr. 14. Úvodní obrazovka po provedení výpočtu

Po provedení výpočtu získáme výstup na Obr. 14. Ve spodní části okna je v příslušných polích zobrazeno řešení úlohy z kapitoly 5. Je zobrazena matice regulátoru  $Q_R$  a součinnic  $F$  a  $P_R$ . Pro výsledný regulátor přitom platí:

$$G_Q(s) = Q_R (FP_R)^{-1}$$

Uživatel si může zvolit jeden ze čtyř způsobů prezentace výstupních hodnot (v okně *Nastavení viz. dále*):

- $Q_R \quad P_R$
- $Q_R \quad FP_R$
- $Q_R \quad (FP_R)^{-1}$
- $Q_R (FP_R)^{-1}$  (kompletní tvar regulátoru Obr. 15)



$$\begin{array}{cc} Qr^*(F*Pr)^{-1} & \\ \frac{(-7s - 3)}{s} & \frac{(2s + 8)}{s} \\ \frac{(4s + 2)}{s} & \frac{(-s - 4)}{s} \end{array}$$

Obr. 15. kompletní tvar regulátoru

#### 6.4.2 Možnosti exportu

Z programu je možno exportovat:

- postup řešení ve formátu txt
- simulační soubor pro prostředí MATLAB

Uživatel může po provedení výpočtu exportovat celý postup řešení do textového souboru. Struktura tohoto souboru je následující. Na začátku jsou vypsány všechny vstupní matice. Následuje zápis postupu řešení. Vždy je popsán příslušný krok řešení a poté vypsána celá struktura matic. Na konec je uveden získaný výsledek. Zápis kroků řešení vzorové úlohy je následující:

$$[\text{sloupec } 4] = [\text{sloupec } 4] + (-2) * [\text{sloupec } 3]$$

$$(-1) * [\text{sloupec } 4]$$

$$[\text{sloupec } 3] = [\text{sloupec } 3] + (-2) * [\text{sloupec } 4]$$

$$[\text{slopec 1}] = [\text{slopec 1}] + (s + 1) * [\text{slopec 3}]$$

$$[\text{slopec 2}] = [\text{slopec 2}] + (s + 4) * [\text{slopec 4}]$$

$$[\text{slopec 1}] = [\text{slopec 1}] + (-2s) * [\text{slopec 4}]$$

Simulační soubor (.m soubor) MATLABu obsahuje vstupní matice a matice regulátoru i realizaci zpětnovazebního řízení pro potřeby simulace. Ta je realizována pomocí funkce *lsim*, která simuluje časovou odezvu dynamického systému na příslušný vstupní signál. Některé parametry simulace je možno změnit v okně nastavení.

### 6.4.3 Okno nastavení

Okno nastavení (Obr. 16) je vyvoláno po kliknutí na volbu *Nastavení* v horním menu. V tomto okně je možno měnit některé parametry aplikace. Změna se projeví po stisknutí tlačítka *OK*. Nové parametry jsou uloženy v uživatelské složce a načteny při příštím spuštění aplikace. Cesta do uživatelské složky je %APPDATA%\MIMO\. Je též možno vyvolat původní nastavení (tlačítko *Původní*).

V tomto okně je možno nastavit:

- Počet zobrazovaných desetinných míst (1-4) - jen pro potřeby výpisu číselných údajů na obrazovku a do textového souboru postupu řešení. Program vnitřně stále pracuje s plnou kapacitou datového typu double. Standardní hodnota: 2.
- Položky ukládané do textového souboru s postupem řešení. Standardně zvoleno vše.
- Parametry souboru MATLABu (vše v sekundách) - parametry pro simulaci v prostředí MATLAB. Jde o délku trvání celé simulace (standardně: 20s), periodu vzorkování (standardně: 0,01s) a počáteční prodlevu před přivedením žádané hodnoty na vstup regulačního obvodu (standardně: 5s).
- Barvy pracovního prostředí (k dispozici je i náhled)
- Tvar zobrazení výstupu (viz. výše)

Nastavení

Počet zobrazovaných desetinných míst  
2

Export pro Matlab

Délka simulace 50

Perioda vzorkování 0,0100

Postup řešení

Ukládat zadání

Ukládat popis kroků řešení

Ukládat v každém kroku celou strukturu

Počáteční prodlevy referenčních signálů

1: 5 2: 5

3: 5 4: 5

5: 5

Barvy

Pozadí Menu Menu Menu

Menu 0123456789

Text 0123456789

Původní barvy

Tvar výstupu

Qr Pr

Qr F\*Pr

Qr (F\*Pr)<sup>-1</sup>

Qr\*(F\*Pr)<sup>-1</sup>

OK Původní Stomo

Obr. 16. Okno nastavení

## 7 UKÁZKOVÉ PŘÍKLADY

Bylo zvoleno několik ukázkových systémů a pro ně navrhnout regulátor pomocí vytvořené aplikace. Tyto výstupy byly porovnány s výsledky řešení stejné úlohy za použití Polynomial Toolboxu. Konkrétně byla použita jeho funkce *axbyc*, která řeší polynomiální maticovou diofantickou rovnici. V každé z ukázkových úloh je úkolem navrhnout spojitý regulátor pro daný systém. Systém je popsán diferenciálními rovnicemi, ze kterých je možno získat popis systému ve tvaru LPMFD (9,10).

### 7.1 Úloha 1

Uvažujme MIMO systém se dvěma vstupy a dvěma výstupy. Tento systém je popsán diferenciálními rovnicemi ve tvaru:

$$\begin{aligned}y_1'(t) + 5y_1(t) + 4y_2(t) &= 0,2u_1(t) + 7u_2(t) \\y_1(t) + 2y_2'(t) + 3y_2(t) &= -1,5u_1(t) + 0,5u_2(t)\end{aligned}$$

Obraz těchto rovnic po Laplaceově transformaci (1):

$$\begin{aligned}sY_1(s) + 5Y_1(s) + 4Y_2(s) &= 0,2U_1(s) + 7U_2(s) \\Y_1(s) + 2sY_2(s) + 3Y_2(s) &= -1,5U_1(s) + 0,5U_2(s)\end{aligned}$$

Popis řízené soustavy ve formě LPMFD:

$$\begin{pmatrix} s+5 & 4 \\ 1 & 2s+3 \end{pmatrix} \begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} = \begin{pmatrix} 0,2 & 7 \\ -1,5 & 0,5 \end{pmatrix} \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix}$$

$$A_L = \begin{pmatrix} s+5 & 4 \\ 1 & 2s+3 \end{pmatrix} \quad B_L = \begin{pmatrix} 0,2 & 7 \\ -1,5 & 0,5 \end{pmatrix}$$

Budeme předpokládat žádané hodnoty a poruchy ve tvaru skoku:

$$w(t) = (h_{w1} \cdot 1(t) \quad h_{w2} \cdot 1(t))^T \quad v(t) = (h_{v1} \cdot 1(t) \quad h_{v2} \cdot 1(t))^T$$

$$w(s) = \begin{pmatrix} \frac{h_{w1}}{s} & \frac{h_{w2}}{s} \end{pmatrix}^T \quad v(s) = \begin{pmatrix} \frac{h_{v1}}{s} & \frac{h_{v2}}{s} \end{pmatrix}^T$$

Tedy matice  $F$  bude ve tvaru:

$$F = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix}$$

Zvolíme dvojnásobné póly  $p_1 = -2$  a  $p_2 = -3$ . Matice požadovaného chování  $D$  tedy bude:

$$D = \begin{pmatrix} (s+2)^2 & 0 \\ 0 & (s+3)^2 \end{pmatrix}$$

Řešení úlohy získané pomocí vytvořené aplikace

$$\tilde{P}_R = \begin{pmatrix} 1 & 0 \\ 0 & 0,5 \end{pmatrix} \quad Q_R = \begin{pmatrix} 0,613s + 0,189 & -3,066s - 5,943 \\ -0,160s + 0,566 & -0,198s + 0,170 \end{pmatrix}$$

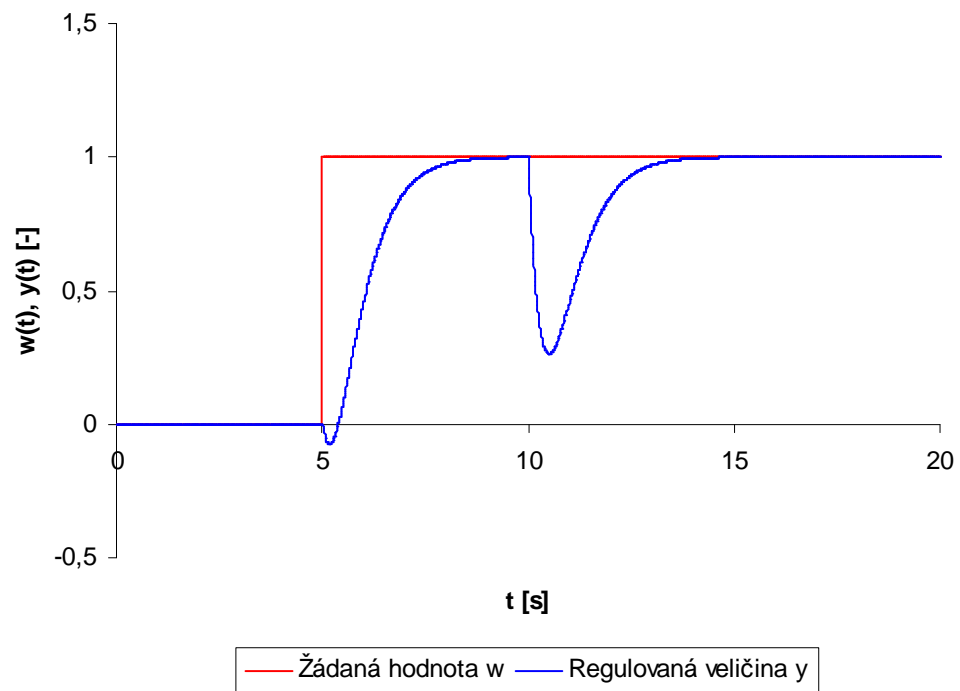
Výsledný regulátor má potom tvar (28):

$$G_Q(s) = \begin{pmatrix} \frac{0,613s + 0,189}{s} & \frac{-3,066s - 5,943}{0,5s} \\ \frac{-0,160s + 0,566}{s} & \frac{-0,198s + 0,170}{0,5s} \end{pmatrix}$$

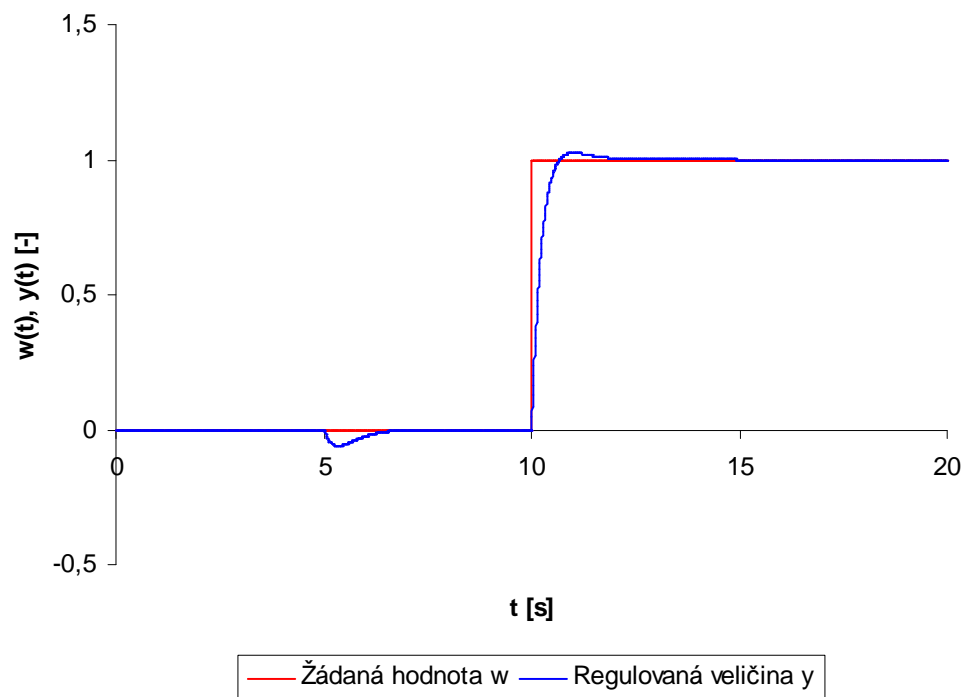
Řešení úlohy získané pomocí Polynomial Toolboxu:

$$\tilde{P}_R = \begin{pmatrix} 1 & 0 \\ 0 & 0,5 \end{pmatrix} \quad Q_R = \begin{pmatrix} 0,61s + 0,19 & -3,1s - 5,9 \\ -0,16s + 0,57 & -0,2s + 0,17 \end{pmatrix}$$

Je patrné, že oba výsledky se od sebe liší jen nepatrně. Rozdíl je nejspíše způsoben odlišným zaokrouhlováním. Příslušné regulační pochody jsou zobrazeny na Obr. 17 a Obr. 18. Žádaná hodnota prvního výstupu je skokově nastavena na hodnotu 1 v čase 5s. Žádaná hodnota druhého výstupu je skokově nastavena na hodnotu 1 v čase 10s. Regulátor pracuje správně. Regulovaná veličina  $y$  dosáhne žádané hodnoty v obou případech v konečném čase s nepatrným překmitem.



Obr. 17. Odezva prvního výstupu systému na skokovou změnu žádané hodnoty



Obr. 18. Odezva druhého výstupu systému na skokovou změnu žádané hodnoty

## 7.2 Úloha 2

Uvažujme MIMO systém se dvěma vstupy a dvěma výstupy. Tento systém je popsán diferenciálními rovnicemi ve tvaru:

$$\begin{aligned}y_1'(t) + 0,5y_1(t) - 3y_2(t) &= -2,3u_1(t) + 5,2u_2(t) \\ 0,1y_1(t) + y_2'(t) - 2y_2(t) &= 0,6u_1(t) + 10u_2(t)\end{aligned}$$

Obraz těchto rovnic po Laplaceově transformaci (1):

$$\begin{aligned}sY_1(s) + 0,5Y_1(s) - 3Y_2(s) &= -2,3U_1(s) + 5,2U_2(s) \\ 0,1Y_1(s) + sY_2(s) - 2Y_2(s) &= 0,6U_1(s) + 10U_2(s)\end{aligned}$$

Popis řízené soustavy ve formě LPMFD:

$$\begin{pmatrix} s + 0,5 & -3 \\ 0,1 & s - 2 \end{pmatrix} \begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} = \begin{pmatrix} -2,3 & 5,2 \\ 0,6 & 10 \end{pmatrix} \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix}$$

$$A_L = \begin{pmatrix} s + 0,5 & -3 \\ 0,1 & s - 2 \end{pmatrix} \quad B_L = \begin{pmatrix} -2,3 & 5,2 \\ 0,6 & 10 \end{pmatrix}$$

Budeme předpokládat žádané hodnoty a poruchy ve tvaru harmonického signálu  $\omega=2$ :

$$w(t) = (h_{w1} \cdot \sin(2t) \quad h_{w2} \cdot \sin(2t))^T \quad v(t) = (h_{v1} \cdot \sin(2t) \quad h_{v2} \cdot \sin(2t))^T$$

$$w(s) = \begin{pmatrix} \frac{4h_{w1}}{s^2 + 4} & \frac{4h_{w2}}{s^2 + 4} \end{pmatrix}^T \quad v(s) = \begin{pmatrix} \frac{4h_{v1}}{s^2 + 4} & \frac{4h_{v2}}{s^2 + 4} \end{pmatrix}^T$$

Tedy matice  $F$  bude ve tvaru:

$$F = \begin{pmatrix} s^2 + 4 & 0 \\ 0 & s^2 + 4 \end{pmatrix}$$

Zvolíme trojnásobné póly  $p_1 = -0,5$  a  $p_2 = -1$ . Matice požadovaného chování  $D$  tedy bude:

$$D = \begin{pmatrix} (s + 0,5)^3 & 0 \\ 0 & (s + 1)^3 \end{pmatrix}$$

Řešení úlohy získané pomocí vytvořené aplikace

$$\tilde{P}_R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad Q_R = \begin{pmatrix} -0,403s^2 + 1,244s + 0,638 & -0,153s^2 - 0,199s - 2,802 \\ 0,014s^2 - 0,075s - 0,078 & 0,509s^2 - 0,088s + 1,068 \end{pmatrix}$$

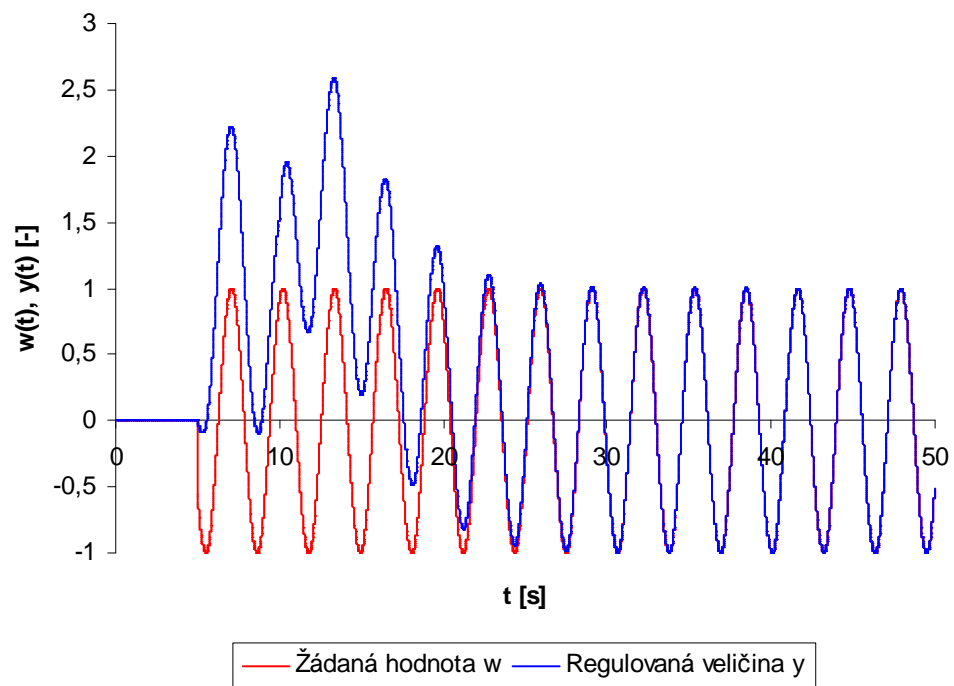
Výsledný regulátor má potom tvar (28):

$$G_Q(s) = \begin{pmatrix} \frac{-0,403s^2 + 1,244s + 0,638}{s^2 + 4} & \frac{-0,153s^2 - 0,199s - 2,802}{s^2 + 4} \\ \frac{0,014s^2 - 0,075s - 0,078}{s^2 + 4} & \frac{0,509s^2 - 0,088s + 1,068}{s^2 + 4} \end{pmatrix}$$

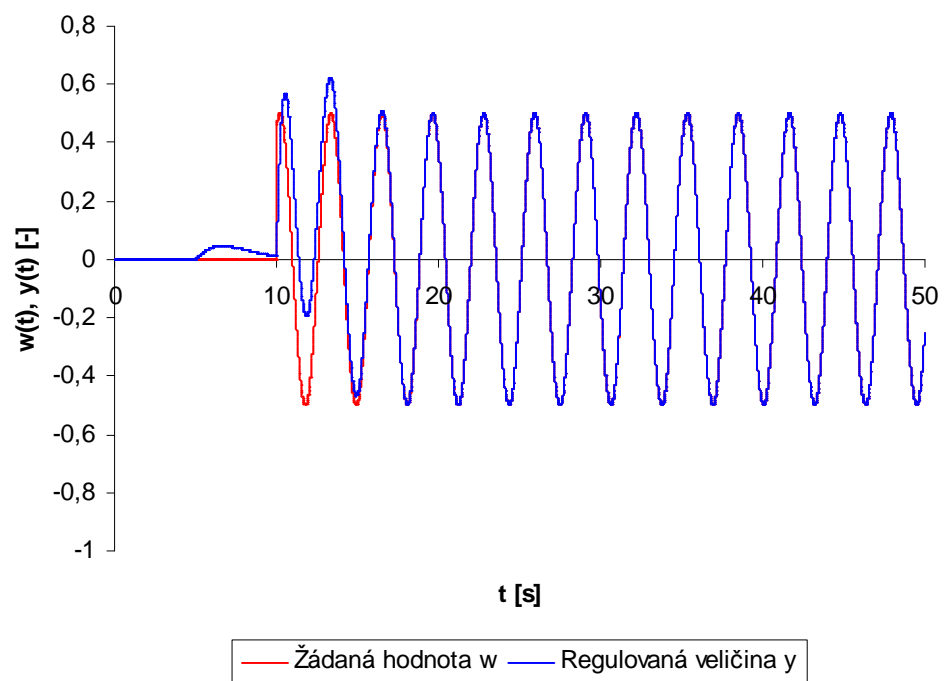
Řešení úlohy získané pomocí Polynomial Toolboxu:

$$\tilde{P}_R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad Q_R = \begin{pmatrix} -0,4s^2 + 1,2s + 0,64 & -0,15s^2 - 0,2s - 2,8 \\ 0,014s^2 - 0,075s - 0,078 & 0,51s^2 - 0,088s + 1,1 \end{pmatrix}$$

Oba výsledky se od sebe opět liší jen nepatrně. Příslušné regulační pochody jsou zobrazeny na Obr. 19 a Obr. 20. Žádaná hodnota prvního výstupu má tvar  $w_1 = \sin(2t)$  od času 5s. Žádaná hodnota druhého výstupu má tvar  $w_2 = 0,5 \sin(2t)$  od času 10s. Regulátor pracuje správně. Regulovaná veličina  $y$  dosáhne žádané hodnoty v obou případech v konečném čase. Lepší chování vykazuje odezva druhého výstupu systému.



Obr. 19. Odezva prvního výstupu systému na harmonickou změnu žádané hodnoty



Obr. 20. Odezva druhého výstupu systému na harmonickou změnu žádané hodnoty

### 7.3 Úloha 3

Uvažujme MIMO systém se dvěma vstupy a dvěma výstupy. Tento systém je popsán diferenciálními rovnicemi ve tvaru:

$$\begin{aligned} 2y_1'(t) + 2y_1(t) + 4y_2(t) &= 6u_1(t) + 2,2u_2(t) \\ -0,5y_1(t) - y_2'(t) + 5y_2(t) &= 9u_1(t) + u_2(t) \end{aligned}$$

Obraz těchto rovnic po Laplaceově transformaci (1):

$$\begin{aligned} 2sY_1(s) + 2Y_1(s) + 4Y_2(s) &= 6U_1(s) + 2,2U_2(s) \\ -0,5Y_1(s) - sY_2(s) + 5Y_2(s) &= 9U_1(s) + U_2(s) \end{aligned}$$

Popis řízené soustavy ve formě LPMFD:

$$\begin{pmatrix} 2s + 2 & 4 \\ -0,5 & -s + 5 \end{pmatrix} \begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} = \begin{pmatrix} 6 & 2,2 \\ 9 & 1 \end{pmatrix} \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix}$$

$$A_L = \begin{pmatrix} 2s + 2 & 4 \\ -0,5 & -s + 5 \end{pmatrix} \quad B_L = \begin{pmatrix} 6 & 2,2 \\ 9 & 1 \end{pmatrix}$$

Budeme předpokládat první žádanou hodnotu ve tvaru harmonického signálu  $\omega=1$ , druhou žádanou hodnotu ve tvaru skoku a poruchy ve tvaru harmonického signálu  $\omega=1$ :

$$w(t) = (h_{w1} \cdot \sin(t) \quad h_{w2} \cdot 1(t))^T \quad v(t) = (h_{v1} \cdot \sin(t) \quad h_{v2} \cdot \sin(t))^T$$

$$w(s) = \left( \frac{h_{w1}}{s^2 + 1} \quad \frac{h_{w2}}{s} \right)^T \quad v(s) = \left( \frac{h_{v1}}{s^2 + 1} \quad \frac{h_{v2}}{s^2 + 1} \right)^T$$

Tedy matice  $F$  bude ve tvaru:

$$F = \begin{pmatrix} s^3 + s & 0 \\ 0 & s^3 + s \end{pmatrix}$$

Zvolíme čtyřnásobné póly  $p_1 = -2$  a  $p_2 = -2$ . Matice požadovaného chování  $D$  tedy bude:

$$D = \begin{pmatrix} (s + 2)^4 & 0 \\ 0 & (s + 2)^4 \end{pmatrix}$$

Řešení úlohy získané pomocí vytvořené aplikace

$$\tilde{P}_R = \begin{pmatrix} 0,5 & 0 \\ 0 & -1 \end{pmatrix}$$

$$Q_R = \begin{pmatrix} -0,467s^3 - 1,667s^2 - 2,207s - 1,159 & 1,783s^3 + 3,667s^2 + 5,609s + 2,551 \\ 4,457s^3 + 15s^2 + 20,109s + 10,435 & -3,043s^3 - 10s^2 - 13,478s - 6,957 \end{pmatrix}$$

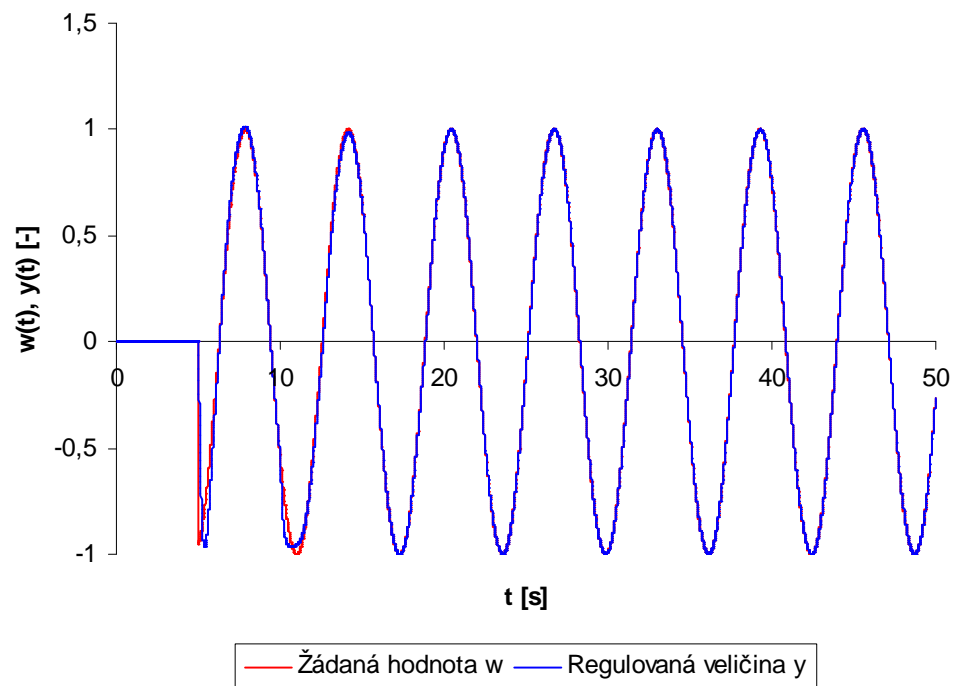
Výsledný regulátor má potom tvar (28):

$$G_Q(s) = \begin{pmatrix} \frac{-0,467s^3 - 1,667s^2 - 2,207s - 1,159}{0,5s^3 + 0,5s} & \frac{1,783s^3 + 3,667s^2 + 5,609s + 2,551}{-s^3 - s} \\ \frac{4,457s^3 + 15s^2 + 20,109s + 10,435}{0,5s^3 + 0,5s} & \frac{-3,043s^3 - 10s^2 - 13,478s - 6,957}{-s^3 - s} \end{pmatrix}$$

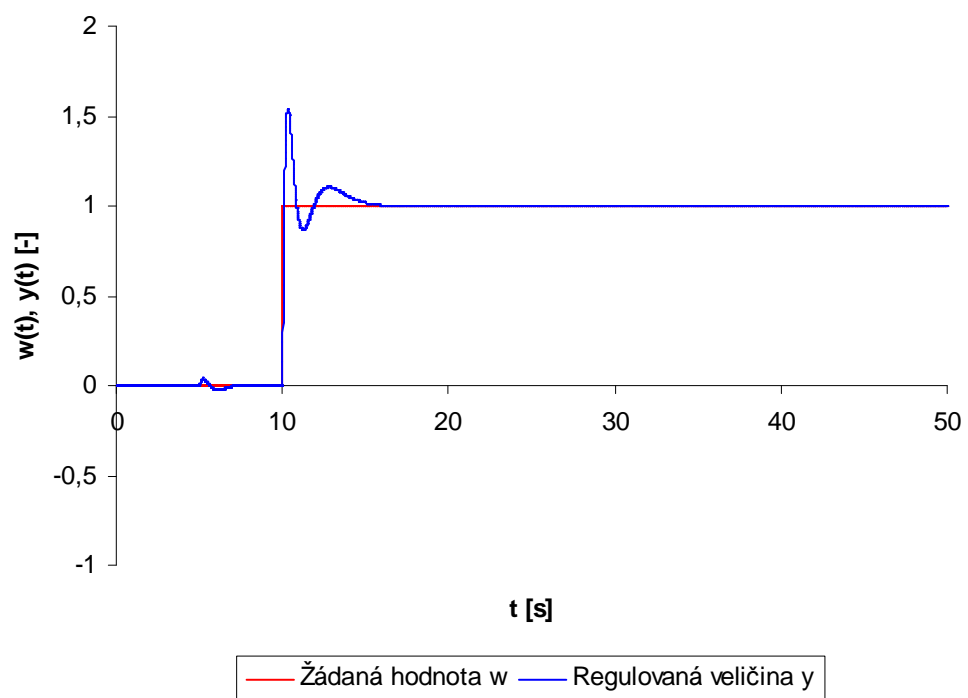
Řešení úlohy získané pomocí Polynomial Toolboxu:

$$\tilde{P}_R = \begin{pmatrix} 0,5 & 0 \\ 0 & -1 \end{pmatrix} \quad Q_R = \begin{pmatrix} -0,47s^3 - 1,7s^2 - 2,2s - 1,2 & 1,8s^3 + 3,7s^2 + 5,6s + 2,6 \\ 4,5s^3 + 15s^2 + 20s + 10 & -3s^3 - 10s^2 - 13s - 7 \end{pmatrix}$$

Oba výsledky se od sebe opět liší jen nepatrně. Příslušné regulační pochody jsou zobrazeny na Obr. 21 a Obr. 22. Žádaná hodnota prvního výstupu má tvar  $w_1 = \sin(t)$  a začíná působit v čase 5s. Žádaná hodnota druhého výstupu je skokově nastavena na hodnotu 1 v čase 10s. Regulátor pracuje správně. Regulovaná veličina  $y$  dosáhne žádané hodnoty v obou případech v konečném čase. Reakce na skokovou změnu žádané hodnoty na druhém výstupu probíhá s překmitem.



Obr. 21. Odezva prvního výstupu systému na harmonickou změnu žádané hodnoty



Obr. 22. Odezva druhého výstupu systému na skokovou změnu žádané hodnoty

## 7.4 Úloha 4

Uvažujme MIMO systém se třemi vstupy a třemi výstupy. Tento systém je popsán diferenciálními rovnicemi ve tvaru:

$$\begin{aligned}y_1'(t) - 0,1y_1(t) - 1,19y_3(t) &= u_1(t) - 2,3u_2(t) \\0,2y_1(t) + y_2'(t) + 5y_2(t) + 4y_3(t) &= -5u_1(t) + 3u_2(t) + 0,8u_3(t) \\0,3y_1(t) + y_3'(t) + 10y_3(t) &= 4,1u_1(t) + u_3(t)\end{aligned}$$

Obraz těchto rovnic po Laplaceově transformaci (1):

$$\begin{aligned}sY_1(s) - 0,1Y_1(s) - 1,19Y_3(s) &= U_1(s) - 2,3U_2(s) \\0,2Y_1(s) + sY_2(s) + 5Y_2(s) + 4Y_3(s) &= -5U_1(s) + 3U_2(s) + 0,8U_3(s) \\0,3Y_1(s) + sY_3(s) + 10Y_3(s) &= 4,1U_1(s) + U_3(s)\end{aligned}$$

Popis řízené soustavy ve formě LPMFD:

$$\begin{pmatrix} s - 0,1 & 0 & -1,19 \\ 0,2 & s + 5 & 4 \\ 0,3 & 0 & s + 10 \end{pmatrix} \begin{pmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{pmatrix} = \begin{pmatrix} 1 & -2,3 & 0 \\ -5 & 3 & 0,8 \\ 4,1 & 0 & 1 \end{pmatrix} \begin{pmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \end{pmatrix}$$

$$A_L = \begin{pmatrix} s - 0,1 & 0 & -1,19 \\ 0,2 & s + 5 & 4 \\ 0,3 & 0 & s + 10 \end{pmatrix} \quad B_L = \begin{pmatrix} 1 & -2,3 & 0 \\ -5 & 3 & 0,8 \\ 4,1 & 0 & 1 \end{pmatrix}$$

Budeme předpokládat žádané hodnoty ve tvaru rampy a poruchy ve tvaru skoku:

$$w(t) = (h_{w1} \cdot t \quad h_{w2} \cdot t \quad h_{w3} \cdot t)^T \quad v(t) = (h_{v1} \cdot 1(t) \quad h_{v2} \cdot 1(t) \quad h_{v3} \cdot 1(t))^T$$

$$w(s) = \left( \frac{h_{w1}}{s^2} \quad \frac{h_{w2}}{s^2} \quad \frac{h_{w3}}{s^2} \right)^T \quad v(s) = \left( \frac{h_{v1}}{s} \quad \frac{h_{v2}}{s} \quad \frac{h_{v3}}{s} \right)^T$$

Tedy matice  $F$  bude ve tvaru:

$$F = \begin{pmatrix} s^2 & 0 \\ 0 & s^2 \end{pmatrix}$$

Zvolíme trojnásobné póly  $p_1 = -1$   $p_2 = -2$  a  $p_3 = -1$ . Matice požadovaného chování  $D$  bude:

$$D = \begin{pmatrix} (s+1)^3 & 0 & 0 \\ 0 & (s+2)^3 & 0 \\ 0 & 0 & (s+1)^3 \end{pmatrix}$$

Řešení úlohy získané pomocí vytvořené aplikace

$$\tilde{P}_R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Q_R = \begin{pmatrix} -0,585s^2 - 0,561s - 0,187 & -0,143s^2 - 1,720s - 1,147 & -0,452s^2 + 0,344s + 0,115 \\ -1,602s^2 - 1,548s - 0,516 & -0,062s^2 - 0,748s - 0,499 & -0,714s^2 + 0,150s + 0,050 \\ 2,100s^2 + 2,300s + 0,767 & 0,588s^2 + 7,053s + 4,702 & -5,147s^2 + 1,589s + 0,530 \end{pmatrix}$$

Výsledný regulátor má potom tvar(28):

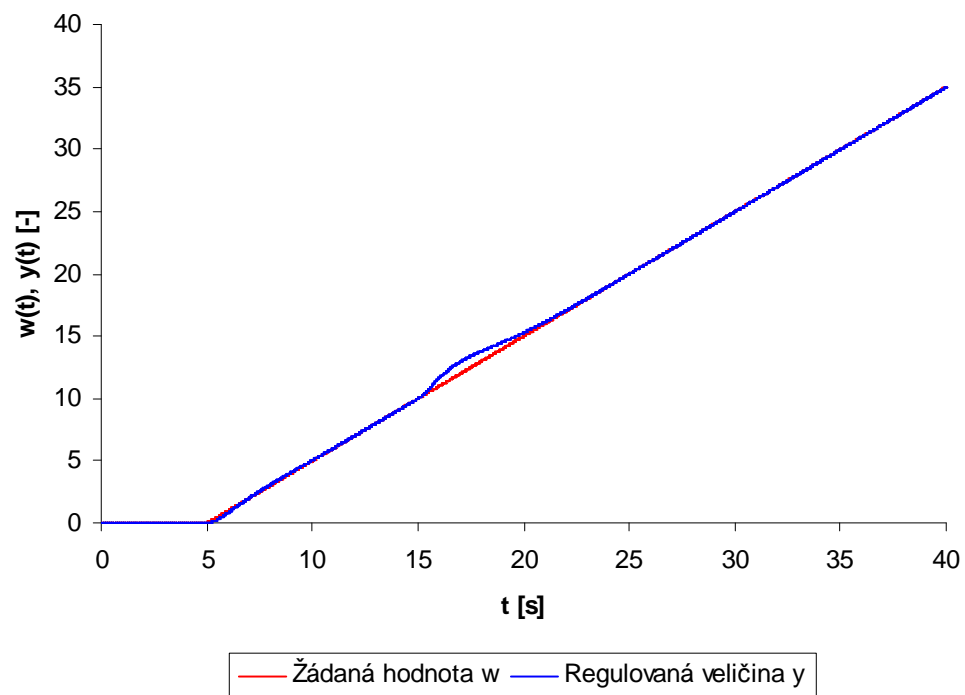
$$G_Q(s) = \begin{pmatrix} \frac{-0,585s^2 - 0,561s - 0,187}{s^2} & \frac{-0,143s^2 - 1,720s - 1,147}{s^2} & \frac{-0,452s^2 + 0,344s + 0,115}{s^2} \\ \frac{-1,602s^2 - 1,548s - 0,516}{s^2} & \frac{-0,062s^2 - 0,748s - 0,499}{s^2} & \frac{-0,714s^2 + 0,150s + 0,050}{s^2} \\ \frac{2,100s^2 + 2,300s + 0,767}{s^2} & \frac{0,588s^2 + 7,053s + 4,702}{s^2} & \frac{-5,147s^2 + 1,589s + 0,530}{s^2} \end{pmatrix}$$

Řešení úlohy získané pomocí Polynomial Toolbox:

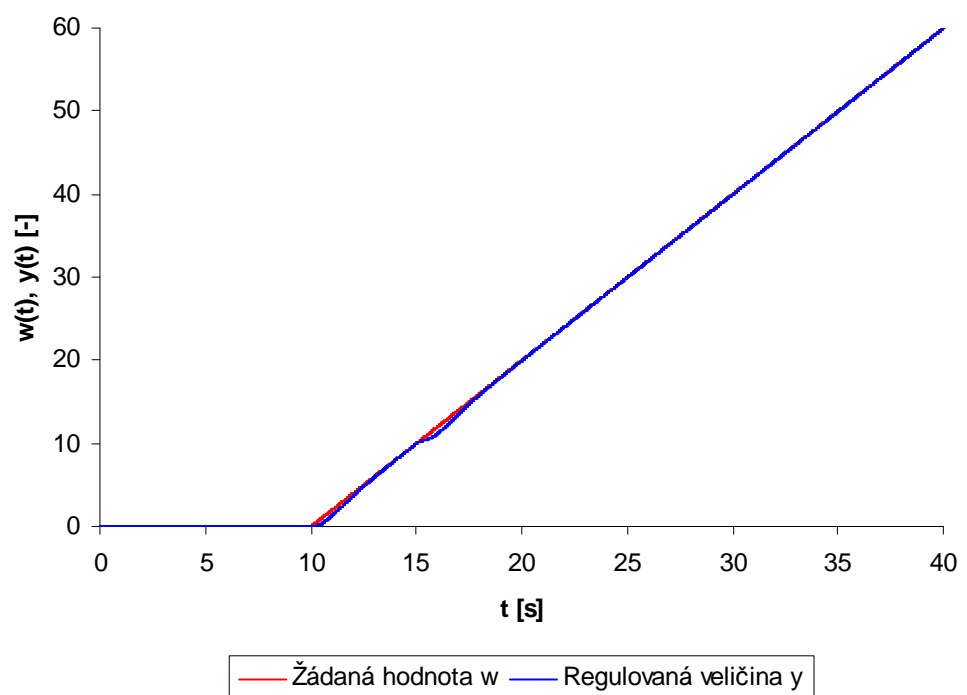
$$\tilde{P}_R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Q_R = \begin{pmatrix} -0,59s^2 - 0,56s - 0,19 & -0,14s^2 - 1,7s - 1,1 & -0,45s^2 + 0,34s + 0,11 \\ -1,6s^2 - 1,5s - 0,52 & -0,062s^2 - 0,75s - 0,5 & -0,71s^2 + 0,15s + 0,05 \\ 2,1s^2 + 2,3s + 0,77 & 0,59s^2 + 7,1s + 4,7 & -5,1s^2 + 1,6s + 0,53 \end{pmatrix}$$

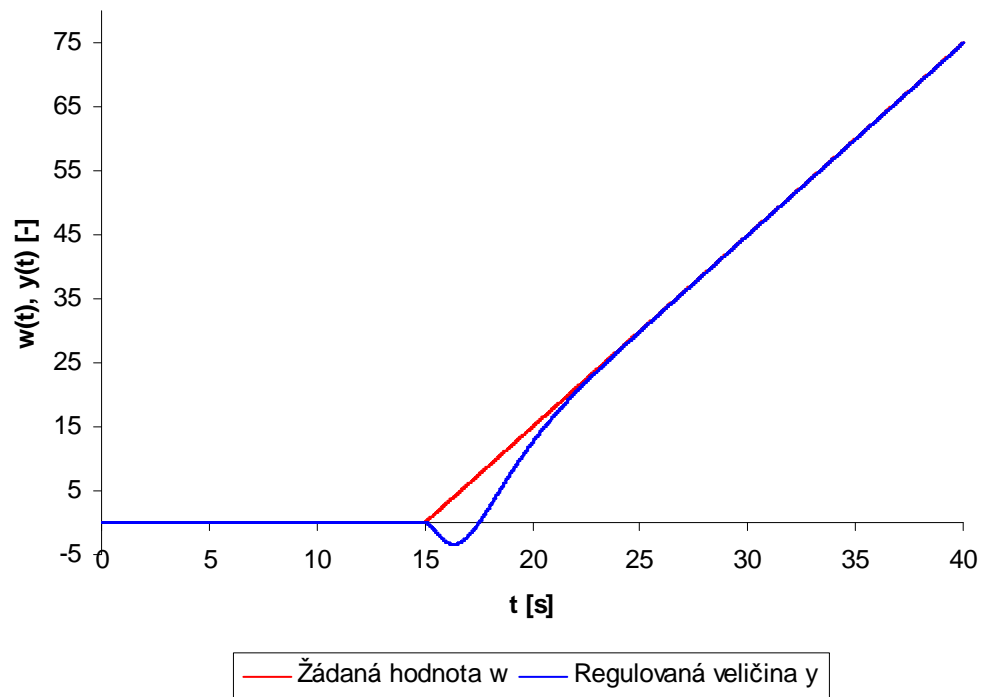
Oba výsledky se od sebe opět liší jen nepatrně. Příslušné regulační pochody jsou zobrazeny na Obr. 23, Obr. 24 a Obr. 25. Žádaná hodnota prvního výstupu má tvar  $w_1 = t - 5$  a začíná působit v časovém okamžiku 5s. Žádaná hodnota druhého výstupu má tvar  $w_2 = 2(t - 10)$  a začíná působit v časovém okamžiku 10s. Žádaná hodnota třetího výstupu má tvar  $w_3 = 3(t - 15)$  a začíná působit v časovém okamžiku 15s. Regulátor pracuje správně. Regulovaná veličina  $y$  dosáhne žádané hodnoty ve všech třech případech v konečném čase.



Obr. 23. Odezva prvního výstupu systému na změnu žádané hodnoty ve tvaru rampy



Obr. 24. Odezva druhého výstupu systému na změnu žádané hodnoty ve tvaru rampy



Obr. 25. Odezva třetího výstupu systému na změnu žádané hodnoty ve tvaru rampy

## ZÁVĚR

Hlavním cílem této diplomové práce bylo vytvoření aplikace pro podporu návrhu MIMO řízení polynomiálním přístupem.

Práce byla rozdělena na teoretickou a praktickou část. V teoretické části byly popsány možnosti realizace takovéto programové podpory. Dále byla teoretická část věnována představení základních pojmů a teoretických principů použité metody. Praktická část práce popisuje použité vývojové prostředky a to programovací jazyk C# a platformu .NET. Popisuje také použitý algoritmus řešení úlohy návrhu regulátoru polynomiálním přístupem. V praktické části práce jsou také prezentovány výsledky návrhu regulátorů pro vybrané úlohy.

Jako výstup práce byla vytvořena programová podpora návrhu MIMO řízení polynomiálním přístupem. Tato aplikace umožňuje navrhnout mnohorozměrný regulátor polynomiálním přístupem s využitím elementárních sloupcových operací na polynomiální matici. Vstupem aplikace jsou polynomiální matice řízeného systému, tvar vstupních signálů, poruch a požadované chování výsledného regulačního obvodu. Výstupem pak je navržený regulátor, který zajistí požadované chování. Ověření výsledku návrhu řízení je realizováno v prostředí MATLAB. Je možno také uložit postup řešení zadané úlohy v textové formě. Na čtyřech úlohách byla demonstrována funkčnost této aplikace a výsledky byly porovnány s řešením stejné úlohy pomocí Polynomial Toolboxu. Využití vytvořené aplikace je možné jak pro praktické, tak i studijní účely.

## ZÁVĚR V ANGLIČTINĚ

The main objective of this thesis was to develop an application for computer-aided polynomial MIMO control system design.

The thesis was divided into theoretical and practical part. The theoretical part describes possibilities of implementation of this computer-aided design. The theoretical part deals with basic concepts and principles of used method. The practical part describes used development tools, programming language C# and .NET platform. It describes used algorithm also. The functionality was demonstrated on several sample tasks.

Application for computer-aided polynomial MIMO control system design was made as the output of this thesis. This application allows user to design regulator for MIMO system using elementary column operations with polynomial matrix. Inputs of the application are polynomial matrices that describe controlled system, input signals, disturbances and behavior of a control loop. The output is the controller, which provides the required behavior. Verification of results is realized in MATLAB. It is possible to save the procedure of solving as a text file. The functionality was demonstrated on four tasks and results were compared with results obtained using Polynomial Toolbox. Created application could be used for practical and learning purposes.

**SEZNAM POUŽITÉ LITERATURY**

- [1] DOSTÁL, Petr; MATUŠŮ, Radek. Stavová a algebraická teorie řízení. Zlín: UTB ve Zlíně, Fakulta aplikované informatiky, 2010. ISBN 978-80-7318-991-4.
- [2] PROKOP, Roman, Radek MATUŠŮ a Zdenka PROKOPOVÁ. Teorie automatického řízení: lineární spojité dynamické systémy. Zlín: UTB ve Zlíně, 2006. ISBN 8073183692.
- [3] HAVLENA, Vladimír a Jan ŠTECHA. Moderní teorie řízení. Praha: ČVUT, 1996. ISBN 8001010767.
- [4] SKOGESTAD, Sigurd; POSTLETHWAITE, Ian. Multivariable Feedback Control: Analysis and Design. Chichester: Wiley, 2005. ISBN 0-470-01168-8.
- [5] BARTSCH, Hans Jochen. *Matematické vzorce*. Praha: Mladá fronta, 2002. ISBN 80-204-0607-7.
- [6] MICROSOFT. *Visual Studio 2010* [online]. 2010 [cit. 2013-05-18]. Dostupné z: <<http://msdn.microsoft.com/query/dev10.query?appId=Dev10IDEF1&l=EN-US&k=k%28MSDNSTART%29&rd=true>>
- [7] BĚHÁLEK, Marek. Programovací jazyk C#. [online]. 2007 [cit. 2013-05-21]. Dostupné z: <<http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/>>
- [8] MATLAB: Jazyk pro technické výpočty. HUMUSOFT. [online]. [cit. 2013-05-21]. Dostupné z: <<http://www.humusoft.cz/produkty/matlab/matlab/index.php?lang=cz&p1=1&p2=1&p3=1>>
- [9] POLYX LTD. The Polynomial Toolbox [online]. [cit. 2013-05-21]. Dostupné z: <<http://www.polyx.com/>>
- [10] KEOGH, Jim a Mario GIANNINI. *OOP bez předchozích znalostí: Průvodce pro samouky*. Brno: Computer Press, a.s., 2006. ISBN 80-251-0973-9.
- [11] ŠEBEK, Michael. *Systémy a řízení: 8. přednáška - Polynomiální metody* [online]. 2007 [cit. 2013-02-04]. ČVUT v Praze. Dostupné z: <<http://www.avc.cvut.cz/archiv/prednasky/prednasky-fel-cvut/x35sri---systemy-a-rizeni>>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

BIBO	Omezený vstup omezený výstup (Bounded-Input Bounded-Output)
CLR	Common Language Runtime
deg	Stupeň
DOF	Stupeň volnosti (Degree of Freedom)
GUI	Grafické uživatelské rozhraní (Graphical User Interface)
LPMFD	Levý polynomiální maticový zlomek (Left Polynomial Matrix Fraction Description)
LSDS	Lineární spojité dynamický systém
MIMO	Multi-Input Multi-Output
NSD	Největší společný dělitel
OOP	Objektově orientované programování
$\mathfrak{R}[s]$	Okruh polynomů
RPMFD	Pravý polynomiální maticový zlomek (Right Polynomial Matrix Fraction Description)

## SEZNAM OBRÁZKŮ

<i>Obr. 1. Schéma mnohorozměrného systému</i> .....	17
<i>Obr. 2. Systém řízení</i> .....	20
<i>Obr. 3. Architektura .NET Framework</i> .....	25
<i>Obr. 4. Microsoft Visual Studio</i> .....	27
<i>Obr. 5. Nástroj pro design GUI formulářů</i> .....	28
<i>Obr. 6. Odezva systému na skokovou změnu žádaných hodnot</i> .....	31
<i>Obr. 7. Vývojový diagram pro tvorbu jednotkové matice</i> .....	33
<i>Obr. 8. Polynomiální matice <math>D</math> na pozici <math>AL\cdot F</math></i> .....	34
<i>Obr. 9. Třída Matrix</i> .....	36
<i>Obr. 10. Úvodní obrazovka</i> .....	38
<i>Obr. 11. Okno pro zadání vstupních dat</i> .....	38
<i>Obr. 12. Zadávání tvaru referenčních signálů</i> .....	39
<i>Obr. 13. Volba prvků matice <math>D(s)</math></i> .....	40
<i>Obr. 14. Úvodní obrazovka po provedení výpočtu</i> .....	40
<i>Obr. 15. kompletní tvar regulátoru</i> .....	41
<i>Obr. 16. Okno nastavení</i> .....	43
<i>Obr. 17. Odezva prvního výstupu systému na skokovou změnu žádané hodnoty</i> .....	46
<i>Obr. 18. Odezva druhého výstupu systému na skokovou změnu žádané hodnoty</i> .....	46
<i>Obr. 19. Odezva prvního výstupu systému na harmonickou změnu žádané hodnoty</i> .....	49
<i>Obr. 20. Odezva druhého výstupu systému na harmonickou změnu žádané hodnoty</i> .....	49
<i>Obr. 21. Odezva prvního výstupu systému na harmonickou změnu žádané hodnoty</i> .....	52
<i>Obr. 22. Odezva druhého výstupu systému na skokovou změnu žádané hodnoty</i> .....	52
<i>Obr. 23. Odezva prvního výstupu systému na změnu žádané hodnoty ve tvaru rampy</i> .....	55
<i>Obr. 24. Odezva druhého výstupu systému na změnu žádané hodnoty ve tvaru rampy</i> .....	55
<i>Obr. 25. Odezva třetího výstupu systému na změnu žádané hodnoty ve tvaru rampy</i> .....	56

## SEZNAM PŘÍLOH

P I      Obsah přiloženého CD

## **PŘÍLOHA P I: OBSAH PŘILOŽENÉHO CD**

Na CD se nachází tři adresáře:

- Diplomová práce - tato diplomová práce ve formátu pdf
- Program - vytvořený program MIMO.exe
- Zdrojové kódy - zdrojové kódy vytvořeného programu